



TESIS - KI142502

**PENDETEKSIAN KETIDAKLENGKAPAN
KEBUTUHAN DENGAN TEKNIK KLASIFIKASI
PADA DOKUMEN SPESIFIKASI KEBUTUHAN
PERANGKAT LUNAK**

SUCI NURFAUZIAH
NRP. 5115201038

DOSEN PEMBIMBING
Daniel Oranova Siahaan, S.Kom, M.Sc, PD.ENG.
Fajar Baskoro, S.Kom, M.T

PROGRAM MAGISTER
BIDANG KEAHLIAN REKAYASA PERANGKAT LUNAK
JURUSAN TEKNIK INFORMATIKA
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017

[Halaman ini sengaja dikosongkan]



TESIS - KI142502

DETECTION OF INCOMPLETENESS USING CLASSIFICATION TECHNIQUE ON SRS

**SUCI NURFAUZIAH
NRP. 5115201038**

**DOSEN PEMBIMBING
Daniel Oranova Siahaan, S.Kom, M.Sc, PD.ENG.
Fajar Baskoro, S.Kom, M.T**

**MASTER PROGRAM
THE EXPERT OF SOFTWARE ENGINEERING
DEPARTMENT OF INFORMATICS
FACULTY OF INFORMATION TECHNOLOGY
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2017**

[Halaman ini sengaja dikosongkan]

LEMBAR PENGESAHAN

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Komputer (M.Kom.)
di
Institut Teknologi Sepuluh Nopember Surabaya


oleh:
Suci Nurfauziah
Nrp. 5115201038

Dengan judul :
PENDETEKSIAN KETIDAKLENGKAPAN KEBUTUHAN DENGAN TEKNIK
KLASIFIKASI PADA DOKUMEN SPESIFIKASI KEBUTUHAN PERANGKAT
LUNAK

Tanggal Ujian : 17-7-2017
Periode Wisuda : 2016 Genap

Disetujui oleh:


Daniel Oranova Siahaan, S.Kom, M.Sc, PD.ENG.
NIP. 197411232006041001


(Pembimbing 1)


Fajar Baskoro, S.Kom, M.T
NIP. 197404031999031002


(Pembimbing 2)

Dr. Ir. Siti Rochimah, M.T
NIP. 196810021994032001


(Penguji 1)

Rizky Januar Akbar, S.Kom, M.Eng.
NIP. 198701032014041001


(Penguji 2)

Adhatus Sholichah, S.Kom, M.Sc
NIP. 198508262015042002


(Penguji 3)

Dekan Fakultas Teknologi Informasi,


Dr. Agus Zainal Arifin, S.Kom, M.Kom.
NIP. 197208091995121001

[Halaman ini sengaja dikosongkan]

PENDETEKSIAN KETIDAKLENGKAPAN KEBUTUHAN DENGAN TEKNIK KLASIFIKASI PADA DOKUMEN SPESIFIKASI KEBUTUHAN PERANGKAT LUNAK

Nama mahasiswa : Suci Nurfauziah
NRP : 5115201038
Pembimbing : Daniel Oranova Siahaan, S.Kom, M.Sc,PD.Eng.
Fajar Baskoro, S.Kom, M.T

ABSTRAK

Rekayasa kebutuhan menghasilkan dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL) dan merupakan tahapan yang kritis pada pengembangan perangkat lunak. Kesalahan yang terjadi pada proses rekayasa kebutuhan akan mempengaruhi ketidakberhasilan produk tersebut. Dokumen SKPL sering kali ditulis dengan bahasa alamiah. Karakteristik dokumen SKPL yang baik adalah benar, tidak rancu, konsisten, dapat diperingkat, dapat diverifikasi, dapat dimodifikasi, dapat ditelusuri, dan lengkap. Pada penelitian ini difokuskan pada kelengkapan. Kualitas spesifikasi kebutuhan bisa dinilai berdasarkan pernyataan kebutuhan atau dokumen kebutuhan. Spesifikasi kebutuhan yang lengkap secara jelas mendefinisikan semua situasi yang dihadapi sistem dan dapat dipahami tanpa melibatkan atau terkait pada kebutuhan lain.

Penelitian ini bertujuan untuk membangun model klasifikasi pendeteksian ketidaklengkapan kebutuhan pada dokumen spesifikasi kebutuhan perangkat lunak yang ditulis dengan bahasa alamiah. Penelitian ini membuat *corpus* kebutuhan yang berisi pernyataan kebutuhan lengkap dan pernyataan kebutuhan tidak lengkap. *Corpus* merupakan kesepakatan hasil pelabelan secara manual oleh tiga orang ahli. Dari *Corpus* akan dilakukan pembangkitan kata kunci, ekstraksi fitur, pembangkitan data buatan, perankingan fitur, dan pembangunan model klasifikasi. Nilai performansi Gwet's AC1 digunakan untuk mengetahui apakah model kerangka kerja yang dibangun dapat diandalkan dan dapat mendeteksi adanya ketidaklengkapan pada dokumen spesifikasi kebutuhan perangkat lunak.

Berdasarkan hasil ujicoba dengan menggunakan kombinasi metode adaboost dan C4.5 diperoleh rata-rata indeks kesepakatan pada level *moderate*. Indeks kesepakatan antara ahli dengan kerangka kerja rata-rata berada pada tingkat *moderate*, ini lebih tinggi bila dibandingkan dengan indeks kesepakatan antar ahli sendiri yang hanya rata-rata pada tingkat *fair*.

Kata kunci: Klasifikasi Teks, Pemrosesan Bahasa Alami, Spesifikasi Kebutuhan Perangkat Lunak, Kelengkapan.

[Halaman ini sengaja dikosongkan]

DETECTION OF INCOMPLETENESS USING CLASSIFICATION TECHNIQUE ON SRS

Name : Suci Nurfauziah
Student Identity Number : 5115201038
Supervisor : Daniel Oranova Siahaan, S.Kom, M.Sc,PD.Eng.
Fajar Baskoro, S.Kom, M.T

ABSTRACT

Software requirements produces Software Requirements Specification (SRS) document and this is a critical stage in Software Development. Errors that occur in the software requirements will affect the failure of the product. SRS often written in natural language. Characteristics of a good SRS is correct, unambiguous, consistent, rank for importance, verifiable, modifiable, traceable dan complete. In this study focused on completeness. The quality requirements specification can be assessed based on the statement or requirements document. Requirement specification is complete that defines precisely all the situations confronting the system and can be understood without related another requirements.

This research purpose to establish a classification model incompleteness detection requirements in software requirements specification document written in natural language. This study makes corpus that contain statements of requirement complete and incomplete. Corpus is agreement of manual labeling by three experts. There will be keyword generation, features extraction, data generation synthesis, feature rank dan building classifier model. Corpus will be used for training and testing the classifier. Gwet's AC1 performance value will be used to determine whether the classifier reliable and detect the presence of incompleteness in SRS.

Based on the result of experiment using combination of method adaboost and C4.5 obtained average of agreement index at moderate level. The index of agreement between an expert with framework is moderate levels, this is higher when compared to an index of agreement between expert alone that are only average at fair levels.

Keywords: Natural Language Processing, Text Classification, Software Requirements Specification, Completeness.

[Halaman ini sengaja dikosongkan]

KATA PENGANTAR



Segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis bisa menyelesaikan Tesis yang berjudul “Pendeteksian Ketidaklengkapan Kebutuhan Dengan Teknik Klasifikasi Pada Dokumen Spesifikasi Kebutuhan Perangkat Lunak” sesuai dengan target waktu yang diharapkan.

Pengerjaan Tesis ini merupakan suatu kesempatan yang sangat berharga bagi penulis untuk belajar memperdalam ilmu pengetahuan. terselesaikannya buku Tesis ini, tidak terlepas dari bantuan dan dukungan semua pihak. Oleh karena itu, penulis ingin menyampaikan rasa terima kasih yang sebesar-besarnya kepada:

1. Allah SWT atas limpahan rahmat-Nya sehingga penulis dapat menyelesaikan Tesis ini dengan baik.
2. Ibu Askanah, Bapak Komari dan Ibu Umroh, selaku orang tua penulis yang selalu mendoakan agar selalu diberikan kelancaran dan kemudahan dalam menyelesaikan Tesis ini. Serta menjadi motivasi terbesar untuk mendapatkan hasil yang terbaik.
3. Imam Machfudhi, selaku suami penulis yang selalu mendokan dan mendukung agar selalu diberikan kelancaran dan kemudahan dalam menyelesaikan Tesis ini.
4. Bapak Daniel Oranova Siahaan, S.Kom, M.Sc, PD.Eng dan Fajar Baskoro, S.Kom, M.T selaku dosen pembimbing yang telah memberikan kepercayaan, motivasi, bimbingan, nasehat, perhatian serta semua bantuan yang telah diberikan kepada penulis dalam menyelesaikan Tesis ini.
5. Ibu Dr. Ir. Siti Rochimah, M.T., Ibu Adhatus Sholichah, S.Kom, M.Sc dan Bapak Rizky Januar Akbar, S.Kom, M.Eng selaku dosen penguji yang telah memberikan bimbingan, saran, arahan, dan koreksi dalam pengerjaan Tesis ini.

6. Bapak Ainul Yaqin, Tandhy Simanjuntak, Devi Karolina, Galuh, dan Hendy, selaku ahli Rekayasa Perangkat Lunak yang telah bersedia membantu penulis dalam melabeli dataset.
7. Bapak Waskitho Wibisono, S.Kom., M.Eng., PhD selaku ketua program pascasarjana Teknik Informatika ITS, selaku dosen wali penulis dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.
8. Mbak Lina, Mas Kunto dan segenap staf Tata Usaha yang telah memberikan segala bantuan dan kemudahan kepada penulis selama menjalani kuliah di Teknik Informatika ITS.
9. Adik penulis, Cicik Retno Wati, serta seluruh keluarga besar yang selalu memberi semangat, doa, dukungan dan hiburan kepada penulis.
10. Teman-teman penulis yang selalu memberikan semangat, dukungan dan hiburan kepada penulis.
11. Rekan-rekan angkatan 2015 Pasca Sarjana Teknik Informatika ITS yang telah menemani dan memberikan bantuan serta motivasi untuk segera menyelesaikan Tesis ini.
12. Juga tidak lupa kepada semua pihak yang belum sempat disebutkan satu per satu disini yang telah membantu terselesaikannya Tesis ini.

Sebagai manusia biasa, penulis menyadari bahwa Tesis ini masih jauh dari kesempurnaan dan memiliki banyak kekurangan. Sehingga dengan segala kerendahan hati, penulis mengharapkan saran dan kritik yang membangun dari pembaca.

Surabaya, Juli 2017

DAFTAR ISI

LEMBAR PENGESAHAN	iv
ABSTRAK	vii
ABSTRACT.....	ix
KATA PENGANTAR	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL.....	xvii
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Perumusan Masalah	3
1.3. Tujuan dan Manfaat Penelitian	3
1.4. Batasan Masalah.....	4
BAB 2 KAJIAN PUSTAKA.....	5
2.1. Dokumen Spesifikasi Kebutuhan Perangkat Lunak.....	5
2.2. Kualitas Kebutuhan.....	5
2.2.1. Kelengkapan Pada Kebutuhan	6
2.3. Pemrosesan Teks pada SKPL	9
2.4. Synthetic Minority Oversampling Technique (SMOTE).....	12
2.5. Information Gain.....	13
2.6. Decision Tree	14
2.6.1. C4.5	14
2.6.2. Adaboost	15
2.7. Gwet's AC1.....	15
BAB 3 METODOLOGI PENELITIAN.....	19

3.1. Studi Literatur.....	19
3.2. Pengembangan Metode.....	20
3.2.1. Pembuatan <i>Corpus</i>	20
3.2.2. Praproses.....	23
3.2.3. Pengembangan Klasifikasi	37
3.3. Analisis Pengujian	38
BAB 4 UJI COBA DAN EVALUASI	40
4.1. Implementasi Penelitian	40
4.2. Perancangan Uji Coba	40
4.2.1. Pembentukan Corpus.....	40
4.2.2. Skenario Uji Coba	41
4.2.3. Implementasi Uji Coba.....	42
4.3. Analisa Hasil	49
4.3.1. Analisa Kesepakatan Antar Ahli dan Ahli dengan Kerangka Kerja ...	49
4.3.2. Analisa Penggunaan Fitur Terhadap Klasifikasi Ketidaklengkapan Kebutuhan.....	50
4.3.3. Analisa Salah Prediksi oleh Kerangka Kerja.....	52
BAB 5 PENUTUP.....	58
5.1. Kesimpulan.....	58
5.2. Saran	58
DAFTAR PUSTAKA.....	60
LAMPIRAN	62
BIODATA PENULIS	81

DAFTAR GAMBAR

Gambar 2.1 Model Taxonomi Kerancuan(Hussain et al. 2007)	10
Gambar 2.2 Pembangkitan Kata Kunci secara Dinamis dan Pemilihan Fitur	11
Gambar 3.1 Tahapan Penelitian	19
Gambar 3.2 Pembuatan <i>Corpus</i>	20
Gambar 3.3 Contoh Data Hasil <i>Corpus</i>	23
Gambar 3.4 Praproses	24
Gambar 3.5 Taxonomi Fitur.....	26
Gambar 3.6 Ekstraksi Fitur	26
Gambar 3.7 Tahapan Eksraksi Kata untuk Kata Kunci Ketidaklengkapan	27
Gambar 3.8 Tahapan Eksraksi Kata dan Perhitungan Frekuensi Fitur Sintaksis..	31
Gambar 3.9 Contoh Data ARFF.....	36
Gambar 3.10 Tahapan Klasifikasi.....	38
Gambar 4.1 Rencana Ujicoba	42
Gambar 4.2 Data format ARFF Hasil Ekstraksi Fitur.....	45
Gambar 4.3 Data Buatan Hasil Pembangkitan Metode SMOTE.....	45
Gambar 4.4 Data Buatan Hasil Pembangkitan Metode SMOTE.....	45
Gambar 4.5 Diagram Distribusi Kesepakatan Ahli.....	49
Gambar 4.6 Grafik Hasil Evaluasi Fitur	51

[Halaman ini sengaja dikosongkan]

DAFTAR TABEL

Tabel 2.1. Tabel Penulisan Hasil Pengamatan	16
Tabel 2.2. Interpretasi Nilai Gwet AC1(Landis & Koch 2012).....	17
Tabel 2.2. Data Pengamatan Contoh Kasus Kappa.....	17
Tabel 3.1 Contoh Kandidat Kata Kunci dari Masing-masing Kategori.....	28
Tabel 3.2 Contoh LR dari kata kunci berkategori POS MD	30
Tabel 3.3 Contoh Hasil Perhitungan Frekuensi Fitur Kata Kunci	30
Tabel 3.4 Penyamaan Tag dengan SentiWordNet	33
Tabel 3.5 Perhitungan Nilai Interpretasi Negatif SentiWordNet	34
Tabel 3.6 Contoh Hasil Perhitungan Frekuensi Fitur Sintaksi.....	35
Tabel 3.7 Contoh Hasil Ekstraksi Fitur.....	35
Tabel 4.1 Tabel Hasil Pembangkitan Kata Kunci	43
Tabel 4.2 Tabel Perankingan Fitur	46
Tabel 4.3 Tabel Hasil Klasifikasi dengan Kombinasi Metode Adaboost dan C4.5	47
Tabel 4.4 Tabel Hasil Klasifikasi dengan Sejumlah Fitur Tertentu.....	48
Tabel 4.5 Tabel Perbandingan Kesepakatan Antar Ahli	50
Tabel 4.6 Tabel Perbandingan Hasil	52
Tabel 4.7 Tabel Pernyataan Kebutuhan yang Beda Kesepakatan Antara Pelabelan Ahli dengan Kerangka Kerja (Menurut Ahli <i>Incomplete</i>)	53
Tabel 4.8 Tabel Pernyataan Kebutuhan yang Beda Kesepakatan Antara Pelabelan Ahli dengan Kerangka Kerja (Menurut Ahli <i>Complete</i>).....	54
Tabel 4.9 Tabel Data Buatan yang Salah Prediksi	55

[Halaman ini sengaja dikosongkan]

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Rekayasa kebutuhan merupakan tahap awal dalam proses pengembangan perangkat lunak, mendeskripsikan sekumpulan kebutuhan yang diperlukan oleh pelanggan dan bagaimana sistem harus berjalan. Rekayasa kebutuhan perangkat lunak berhubungan atau mempengaruhi proses pengembangan perangkat lunak.

Salah satu dokumen yang dihasilkan oleh rekayasa kebutuhan merupakan dokumen Spesifikasi Kebutuhan Perangkat Lunak (SKPL). Rekayasa kebutuhan perangkat lunak merupakan tahapan yang kritis pada siklus SDLC (*Software Development Live Cycle*), maka dalam pembuatan dokumen SKPL haruslah berhati-hati agar menghasilkan sebuah perangkat lunak yang berkualitas. Kesalahan yang terjadi pada rekayasa kebutuhan akan mempengaruhi ketidakberhasilan produk tersebut. Tidak peduli seberapa baik tahapan selanjutnya (Pressman 2012).

Dokumen kebutuhan sering kali ditulis dengan bahasa alamiah karena bersifat intuitif, ekspresif dan universal. Akan tetapi, juga memiliki kelemahan yaitu mudah menimbulkan kerancuan, tidak jelas dan pemaknaannya bergantung pada latar belakang pembaca. Kesalahan seperti ketidakkonsistenan, kerancuan serta ketidaklengkapan pada SKPL diketahui pada awal tahap analisa kebutuhan akan mengurangi biaya yang ditimbulkan akibat error atau cacat pada perangkat lunak yang tanpa adanya pengukuran kualitas kebutuhan.

Penelitian pada bidang rekayasa kebutuhan khususnya tentang mendeteksi kualitas kebutuhan telah banyak dilakukan, beberapa jurnal menjelaskan tentang kerancuan seperti yang dilakukan oleh (Hussain et al. 2007), (Kiyavitskaya et al. 2008) dan (Popescu et al. 2008), *inconsistency* oleh (de Sousa et al. 2010), *conflicts* dilakukan oleh (Sardinha et al. 2013), dan *correctness* oleh (Parra et al. 2015). Pada penelitian ini akan difokuskan untuk

mengklasifikasikan kebutuhan dari dokumen kebutuhan perangkat lunak berdasarkan kelengkapannya.

Spesifikasi kebutuhan yang lengkap adalah spesifikasi kebutuhan yang secara jelas mendefinisikan semua situasi yang dihadapi system (Vie 2010). Kualitas spesifikasi kebutuhan bisa dinilai berdasarkan pernyataan kebutuhan atau dokumen kebutuhan.

Contoh pernyataan dokumen kebutuhan yang lengkap dan tidak lengkap:

- 1) The Flight Information System shall usually be on line.
- 2) The Flight Information System shall have an availability of at least xx% over a period of at least yy hours.

Pada contoh 1) merupakan contoh pernyataan kebutuhan tidak lengkap karena terdapat bentuk ekspresi subyektif “*usually*” dan seharusnya tidak digunakan pada spesifikasi kebutuhan. Pada contoh 2) merupakan bentuk pernyataan kebutuhan yang lengkap dengan mengganti kata *usually* yang dijelaskan lebih rinci (INCOSE 2015).

Beberapa penelitian (Gralha et al. 2015; Genova et al. 2013) melakukan evaluasi kualitas dokumen kebutuhan tentang kelengkapan. Catarina Gralha mengukur kompleksitas dan kelengkapan dokumen kebutuhan yang ditulis dengan i* model (Gralha et al. 2015). I* model merupakan salah satu bentuk bahasa formal sedangkan dokumen SKPL sering ditulis dengan bahasa alamiah. Genova menyebutkan beberapa karakteristik untuk menilai kualitas kebutuhan yang ditulis dengan bahasa alamiah, salah satunya adalah kelengkapan. Beberapa fitur yang diusulkan oleh Genova untuk menilai kelengkapan bersifat tidak langsung (secara independen tidak dapat menentukan kebutuhan yang ditulis tersebut lengkap atau tidak) (Genova et al. 2013). Sehingga pada penelitian ini mendeteksi ketidaklengkapan pernyataan kebutuhan pada dokumen spesifikasi perangkat lunak yang ditulis dengan bahasa alamiah.

Dokumen berbasis teks seperti SKPL dapat dianalisis menggunakan *text mining* atau lebih dikenal dengan *Knowledge Discovery from Text* (KDT). *Text mining* atau KDT menggunakan teknik dari *information retrieval*,

information extraction serta *Natural Language Processing* (NLP) dan menghubungkannya dengan algoritma dan metode pada KDD, *data mining*, *machine learning* dan statistika (Hotho et al. 2005). Salah satu penelitian yang memanfaatkan *text mining* dalam memproses dokumen kebutuhan dilakukan oleh Ishar Husain (Hussain et al. 2007). Dalam penelitian tersebut, dibangun sebuah *classifier* yang dapat mendeteksi adanya kerancuan pada dokumen SKPL. Penelitian tersebut juga menyatakan ketidakadaan standar *corpus* untuk keperluan *training* dan *testing* bagi *classifier* yang dibuat sehingga mereka membuat *corpus* sendiri. Penelitian ini berusaha mengembangkan teknik yang digunakan oleh Hussain untuk membangun model klasifikasi untuk mendeteksi ketidaklengkapan dalam pernyataan kebutuhan perangkat yang ditulis dengan bahasa alamiah. Dalam penelitian ini juga akan dibuat *corpus* untuk keperluan pengklasifikasian ketidaklengkapan.

1.2. Perumusan Masalah

Permasalahan dalam penelitian ini adalah:

1. Bagaimana membuat *corpus* sebagai bahan *training* dan *testing* bagi *classifier* yang akan dibuat?
2. Fitur pernyataan kebutuhan apa saja yang dapat mengindikasikan ada atau tidaknya ketidaklengkapan kebutuhan pada dokumen spesifikasi kebutuhan perangkat lunak dengan teknik klasifikasi?
3. Bagaimana membangun klasifikasi ketidaklengkapan berdasarkan fitur-fitur pernyataan kebutuhan yang telah ditentukan ?

1.3. Tujuan dan Manfaat Penelitian

Tujuan dari penelitian ini adalah mendeteksi ketidaklengkapan kebutuhan dalam dokumen spesifikasi kebutuhan perangkat lunak. Adapun manfaat penelitian ini adalah membantu perekrut kebutuhan perangkat lunak membuat dokumen spesifikasi kebutuhan yang lengkap. Dalam penelitian ini, kontribusi yang diberikan adalah adanya model komputasi untuk mendeteksi ketidaklengkapan pada dokumen kebutuhan perangkat lunak yang ditulis dengan bahasa alamiah.

1.4. Batasan Masalah

Batasan masalah yang menjadi obyek dalam penelitian ini adalah

1. Pernyataan kebutuhan dinyatakan dalam bahasa alamiah dan dalam bahasa Inggris.
2. Indikator kualitas spesifikasi kebutuhan dibatasi pada kelengkapan dari suatu pernyataan kebutuhan.

BAB 2

KAJIAN PUSTAKA

Bab ini menguraikan gambaran tentang kelengkapan kebutuhan, penelitian-penelitian terkini yang terkait dengan deteksi ketidaklengkapan kebutuhan, dan dasar teori yang digunakan untuk membangun solusi yang diajukan. Dasar teori yang digunakan antara lain tentang dokumen spesifikasi kebutuhan perangkat lunak, kualitas kebutuhan, pemrosesan teks pada SKPL, decision tree, dan Gwet's AC1.

2.1. Dokumen Spesifikasi Kebutuhan Perangkat Lunak

Salah satu aktivitas yang dilakukan ketika merekayasa kebutuhan adalah spesifikasi kebutuhan. Spesifikasi kebutuhan merupakan suatu proses memformalisasikan sekumpulan kebutuhan yaitu kebutuhan fungsional maupun non-fungsional dari suatu system yang hendak dibangun ke dalam suatu dokumen.

Dokumen spesifikasi kebutuhan menjelaskan sekumpulan layanan yang diperlukan oleh pengguna dan digunakan oleh pihak pengembang sistem. Spesifikasi kebutuhan perangkat lunak bisa ditulis dengan beberapa metode yaitu bahasa alamiah (*natural language*), bahasa alamiah terstruktur (*structured natural language*), bahasa semi-formal dan bahasa formal (Siahaan 2012).

Bahasa alamiah lebih mudah dipahami oleh *stakeholder*. Bahasa alamiah lebih menguntungkan karena lebih fleksibel untuk menampung deskripsi kebutuhan yang berubah-ubah. Akan tetapi juga memiliki kelemahan, yaitu mudah menimbulkan kerancuan dan susah dianalisa secara otomatis.

2.2. Kualitas Kebutuhan

Dokumen SKPL harus berkualitas untuk menghindari kemungkinan kegagalan perangkat lunak seperti pada tahap pemrograman dan pengujian. Kualitas perangkat lunak bergantung pada tahap rekayasa kebutuhan (Carlson & Laplante 2014).

Penggunaan bahasa alamiah pada spesifikasi kebutuhan sangatlah sulit untuk menjelaskan secara jelas, tepat serta terhindar dari ambigu. Akan tetapi metode ini tetap sering dipergunakan untuk menjelaskan segala sesuatu yang diperlukan oleh sistem. Kebutuhan yang ditulis dengan bahasa alamiah memberi keuntungan diantaranya tidak ada batasan konsep dalam menjelaskan kebutuhan yang diperlukan (INCOSE 2015).

Menurut (Fabbrini et al. 2000) untuk mengevaluasi kebutuhan, maka terdapat dua aspek kualitas SKPL, yaitu :

- Kualitas pernyataan kebutuhan yaitu kualitas dari kalimat tunggal yang dinilai secara terpisah.
- Kualitas dokumen kebutuhan yaitu kualitas dari beberapa kalimat dipertimbangkan dalam konteks dokumen kebutuhan secara keseluruhan.

Pada (ISO/IEC/ IEEE 29148 2011) menjelaskan bahwa sebuah kebutuhan adalah pernyataan yang menjelaskan keperluan dan kendala serta kondisi yang terkait. Sebuah kebutuhan yang ditulis dengan bahasa alamiah harus terdiri dari subjek, predikat, dan pelengkap. Kalimat harus menyatakan subjek dari kebutuhan dan apa yang harus dilakukan. Sedangkan dokumen kebutuhan terdiri dari kumpulan kebutuhan yang sesuai keinginan dari *stakeholder*.

2.2.1. Kelengkapan Pada Kebutuhan

Tidak mudah untuk membuat bahasa alamiah mudah dipahami oleh semua pihak yang berkepentingan pada pembuatan perangkat lunak. Banyak metodologi dan standar sebagai acuan penulisan dokumen spesifikasi kebutuhan yang baik, masing-masing standar terdiri dari beberapa karakteristik kebutuhan.

Berdasarkan IEEE Std 830-1998 karakteristik kebutuhan meliputi tepat, tidak rancu, lengkap, konsisten, dirangking berdasarkan kepentingan, dapat diverifikasi, dapat dimodifikasi, dan dapat dilacak (Siahaan 2012). NASA juga mengeluarkan 10 karakteristik yang harus dimiliki oleh dokumen spesifikasi kebutuhan yaitu lengkap, konsisten, tepat, dapat dimodifikasi, dirangking berdasarkan kepentingan, dapat diukur, dapat dilacak, tidak rancu,

benar, dan dapat diverifikasi (Vie 2010). Berdasarkan ISO/IEC/IEEE 29148, Requirements Working Group dari organisasi INCOSE menulis *Guide for Writing Requirements* yang menjelaskan terdapat dua macam kualitas kebutuhan yaitu kualitas pernyataan kebutuhan (*individual requirements*) dan kualitas dokumen kebutuhan (*set of requirements*). Karakteristik pernyataan kebutuhan meliputi diperlukan, *implementation independent*, tidak rancu, lengkap, tunggal, dapat dikerjakan, dapat diverifikasi, tepat, dan dapat disesuaikan. Sedangkan karakteristik dokumen kebutuhan meliputi lengkap, konsisten, dapat dikerjakan, terbatas ruang lingkup, dan terstruktur (INCOSE 2015).

Contoh pernyataan kebutuhan yang lengkap dan tidak lengkap:

- 1) The Flight Information System shall usually be on line.
- 2) The Flight Information System shall have an availability of at least xx% over a period of at least yy hours.

Pada contoh 1) merupakan contoh pernyataan kebutuhan tidak lengkap karena terdapat bentuk ekspresi subjektif “*usually*” dan seharusnya tidak digunakan pada spesifikasi kebutuhan. Pada contoh 2) merupakan bentuk pernyataan kebutuhan yang lengkap dengan mengganti kata *usually* dan dijelaskan secara rinci (INCOSE 2015).

Sebuah kebutuhan yang lengkap menunjukkan setiap pernyataan kebutuhan dapat dipahami tanpa melibatkan atau terkait pada kebutuhan lain. Pernyataan kebutuhan seharusnya kalimat lengkap yang tidak memerlukan referensi pada pernyataan lain untuk dipahami sebagai bentuk dasar. Sedangkan dokumen kebutuhan dinyatakan lengkap adalah semua fungsi, fitur didefinisikan secara jelas sesuai dengan keinginan stakeholder (INCOSE 2015).

Sebuah kebutuhan juga dikatakan lengkap jika semua informasi terhindar dari kerancuan dan tidak perlu penjelasan tambahan (Firesmith 2005). Pada penelitian (Fabbrini et al. 2000) juga mendefinisikan kelengkapan sebagai salah satu karakteristik pada kualitas pernyataan kebutuhan dan dokumen kebutuhan. Pernyataan kebutuhan lengkap merupakan kemampuan setiap kebutuhan untuk menjelaskan dengan tepat kebutuhannya. Sedangkan

dokumen kebutuhan yang lengkap adalah dokumen spesifikasi kebutuhan yang terhindar dari kemungkinan atau ketidakcocokan diantara pernyataan kebutuhan.

Pada penelitian (Genova et al. 2013) menyebutkan beberapa fitur untuk menilai kualitas kebutuhan yang berpengaruh secara tidak langsung terhadap kelengkapan. Fitur tersebut antara lain :

a. Berdasarkan *morfologi* :

- Ukuran terkait dengan jumlah kata pada pernyataan kebutuhan.

b. Berdasarkan *lexical* :

- Kata sambung meliputi bentuk kata penghubung setara (*and/or*), kata sambung menunjukkan bentuk negative (*not, no, neither, nothing, nowhere, dll*), kata sambung yang menunjukkan klausa atau bentuk aliran (*if...then*), kata sambung yang menunjukkan kata ganti seperti kata ganti personal (*it, they*), *demonstrative pronouns* (*that, those, these, this*)
- Kata rancu/tidak jelas meliputi bentuk ekspresi subjective seperti kata sifat (*good, adequate, efficient, enough, sufficient, dll*) serta kata keterangan (*usually, approximately, sufficiently, typically, dll*).

c. Berdasarkan *analytical* :

- *Verbal tense and mood* meliputi aturan sebagai berikut :
 - ✓ Kebutuhan seharusnya memiliki minimal kata yang menunjukkan penegasan (*imperative verb*). Kata penegasan dalam bahasa inggris dapat ditunjukkan dengan bentuk kata kerja dasar yang didahului oleh kata pelengkap (*must, have to, has to dan had to*)
 - ✓ Menghindari penggunaan kata kerja secara implisit, maksudnya kata kerja yang menunjukkan ekspresi kondisional dalam kalimat. Ekspresi kondisional dengan menerapkan kata-kata *modal* seperti *should, can, may*, kecuali kata *must* yang identic dengan penegasan.
 - ✓ Menghindari penggunaan kata kerja pasif.

- *Domain terms* : Penggunaan kata kerja dan kata benda dalam satu kalimat secara berlebihan.
- d. Berdasarkan *relational* :
- Ketergantungan antar kebutuhan bertujuan untuk mempermudah perbaikan dan penggunaan kembali.
 - *Overlapping* pada seluruh kebutuhan : adanya dua konflik antara dua kebutuhan dan *redundant* pada seperangkat kebutuhan.

Fitur pada point ke a sampai c merupakan fitur kualitas untuk setiap pernyataan kebutuhan, sedangkan fitur pada poin ke d merupakan fitur kualitas untuk dokumen kebutuhan (Genova et al. 2013).

Menurut (INCOSE 2015) yang juga menyebutkan beberapa fitur kelengkapan. Kelengkapan pada setiap pernyataan kebutuhan meliputi semua nomer harus memiliki jenis ukuran secara eksplisit, menghindari kata keterangan, menghindari kata sifat, tidak ada klausa (*no escape clause*), menghindari klausa terbuka (kalimat terdiri dari beberapa kebutuhan yang disebutkan dalam satu kalimat), menghindari penggunaan kata ganti, serta menghindari penggunaan *heading* untuk mendukung penjelasan pernyataan kebutuhan. Sedangkan penilaian kelengkapan pada dokumen kebutuhan adalah kebutuhan harus dapat dilacak.

2.3. Pemrosesan Teks pada SKPL

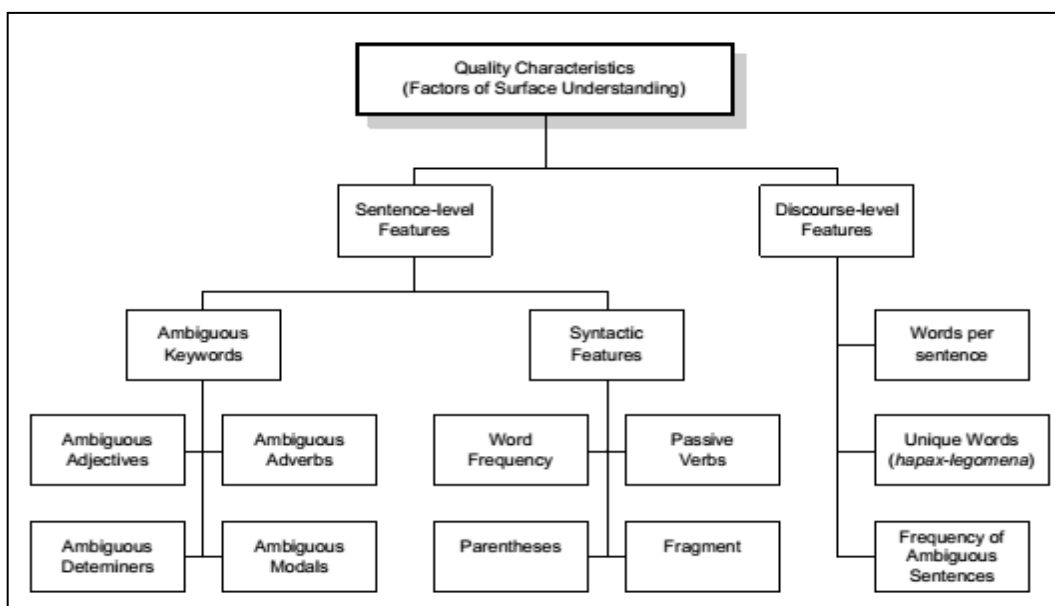
Terdapat beberapa penelitian (Hussain et al. 2007; Parra et al. 2015) yang telah dilakukan terkait pemrosesan text pada SKPL, khususnya pada penilaian kualitas kebutuhan.

1. Ishrar Hussain, 2007

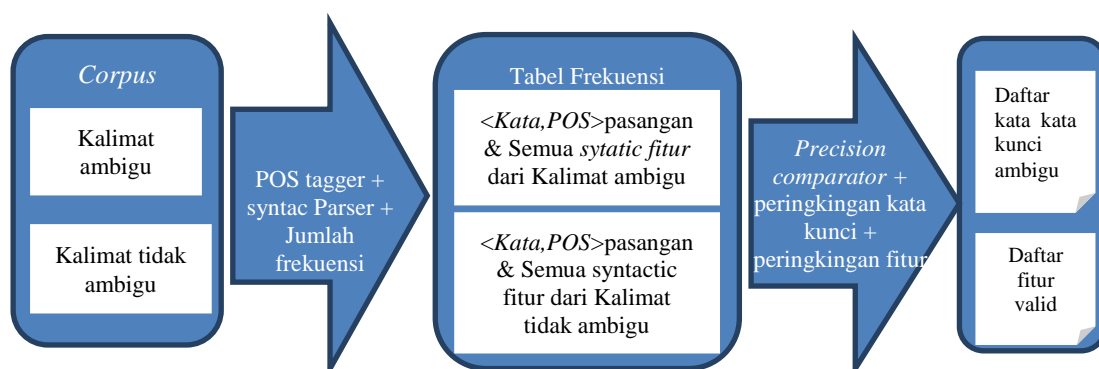
Ishair Hussain pada penelitiannya berhasil membuat klasifikasi untuk mendeteksi kerancuan pada dokumen SKPL yang ditulis dengan bahasa alamiah. Dalam penelitian tersebut Hussain menggunakan *corpus* yang berisi kalimat-kalimat yang sudah ditandai sebagai kalimat ambigu atau tidak ambigu oleh ahli untuk digunakan sebagai data latih dan data uji. Gambar 2.1

merupakan taxonomi yang diusulkan oleh Hussain untuk mengklasifikasikan fitur-fitur yang mungkin mengindikasikan keambiguan (Hussain et al. 2007)

Hussain menggunakan *Stanford Parser* (dilengkapi *Brill's POS tagger* dan *stemmer*) untuk mengekstraksi daftar *syntactic feature* dari setiap kalimat. Selanjutnya dilakukan pembangkitan kata kunci yang ambigu secara dinamis dengan membandingkan *Likelihood Ratio* dari kata kunci kategori POS dan pemilihan fitur yang valid dengan meranking LR dari semua fitur (Hussain 2007).



Gambar 2.1 Model Taxonomi Kerancuan(Hussain et al. 2007)



Gambar 2.2 Pembangkitan Kata Kunci secara Dinamis dan Pemilihan Fitur

Gambar 2.2 mengilustrasikan proses pembangkitan kata kunci secara dinamis dan pemilihan fitur oleh (Hussain et al. 2007). Pembangkitan kata kunci kata kunci ambigu secara dinamis dilakukan dengan cara setiap kata akan dilakukan POS tagging. Selanjutnya kata akan dicatat jumlahnya dan disimpan di tabel frekuensi. Kata akan dihitung rasio frekuensi kemunculannya pada *corpus ambiguity* dan rasio frekuensi kemunculannya pada semua *corpus* untuk mendapatkan rangking kata kunci. Untuk setiap kata kunci rasio ini disebut “*Likelihood Ratio*” (LR), menunjukkan kekuatan diskriminasinya. Kemudian dilakukan peringkingan LR dari kata kunci pada setiap kategori POS secara terpisah, selanjutnya ditentukan nilai *cut-off threshold* untuk setiap kategori POS. LR yang memiliki nilai lebih tinggi dari *threshold* disimpan pada list sebagai *bad_JJ*, *bad_RB*, *bad_MD*, *bad_DT*, sesuai dengan nama POSnya.

Kakas *Feature Extractor* menghitung frekuensi kemunculan dari fitur dan kata kunci ambigu pada setiap kalimat dan menyimpannya sebagai data latih. Data latih juga menyimpan anotasi yang diberikan oleh ahli. *Sentence classifier* dilatih menggunakan data latih untuk mendeteksi ambigu pada data uji secara otomatis dengan mengklasifikasikan kalimat yang belum terklasifikasi ke dalam dua kategori: “*ambiguous*” dan “*unambiguous sentence*”.

Alasan algoritma *Decision Tree C4.5* dipilih oleh Hussain untuk melakukan klasifikasi yaitu : (1) adanya *backtracking* dari tiap node untuk mengetahui alasan dari pengklasifikasian, serta C4.5 (revisi 8) dengan fitur post-pruning merupakan algoritma pembelajaran yang open source terbaik yang diperoleh. (2) ukuran *corpus* tidak cukup besar untuk *training* algoritma *neural network* agar mendapatkan hasil yang lebih baik. Kakas bantu yang dipergunakan untuk klasifikasi yaitu Weka.

2. Parra, 2015

Penelitian lain yang terkait dengan menentukan kualitas kebutuhan secara otomatis juga dilakukan oleh (Parra et al. 2015), pada penelitiannya

menggunakan karakteristik metrik untuk kebenaran. Untuk melakukan ekstraksi fitur, Parra menggunakan kakas bantu yaitu *Requirements Quality Analyzer* (RQA). Parra juga mengusulkan dua model pendekatan untuk tahap *learning instances* atau proses pembentukan *classifier* yaitu model yang pertama disebut dengan “*simple model*” dan model kedua disebut “*model with pairs of comparison*”. Model pertama, satu set *instance* terdiri dari nilai nominal metric dan kualitas yang sudah ditandai oleh ahli yang berasal dari satu kebutuhan yang sama. Model kedua, satu set *instance* terdiri dari gabungan dua kebutuhan. Algoritma klasifikasi yang dipergunakan antara lain C4.5, Part, Bagging dan Boosting. Pembentukan klasifikasi menggunakan kakas bantu Weka.

2.4. Synthetic Minority Oversampling Technique (SMOTE)

Ketidakseimbangan data terjadi apabila jumlah objek dari suatu kelas lebih sedikit dibandingkan dengan kelas lain. Kelas dengan data lebih sedikit disebut kelas minor sedangkan kelas dengan data lebih banyak disebut kelas mayor. Penggunaan data yang tidak seimbang berpengaruh pada hasil model yang diperoleh. Pengolahan algoritma yang tidak menghiraukan keseimbangan data akan cenderung diliputi kelas mayor daripada kelas minor (Chawla et al. 2002).

Metode SMOTE diusulkan oleh (Chawla et al. 2002) sebagai salah satu metode untuk menangani ketidakseimbangan data. Metode SMOTE menambah jumlah data dari kelas minor agar setara dengan jumlah data dari kelas mayor dengan cara membangkitkan data buatan. Pembangkitan data oleh metode SMOTE berdasarkan dari k tetangga terdekat. Pembangkitan data bertipe numerik berbeda dengan bertipe kategorik. Jarak kedekatan data bertipe numerik diukur dengan menggunakan jarak Euclidean sedangkan data bertipe kategorik dengan nilai modus. Berikut tahapan pembangkitan data buatan, untuk (Barro et al. 2013):

- 1) Bertipe numerik :
 - a. Menghitung perbedaan antara vector utama dengan k tetangga terdekat.

- b. Mengalikan perbedaan dengan angka yang diacak antara 0 sampai 1.
 - c. Menambahkan perbedaan tersebut ke dalam nilai utama pada vektor asal sehingga diperoleh vektor utama baru.
- 2) Bertipe kategorik :
- a. Memilih mayoritas pada vektor utama yang dipertimbangkan dengan k-tetangga terdekatnya. Jika terdapat nilai yang sama maka pilih secara acak.
 - b. Tetapkan nilai itu ke sampel kelas minoritas sintetis yang baru.

2.5. Information Gain

Tujuan dari perangkikan fitur dan seleksi fitur untuk mengidentifikasi fitur yang mungkin relevan dan meningkatkan hasil klasifikasi. Seleksi fitur merupakan tahapan praproses klasifikasi. *Information gain* merupakan metode fitur seleksi yang paling populer. *Information gain* melihat fitur secara terpisah dan mengukur betapa pentingnya fitur tersebut dengan meranking untuk prediksi kelas. Pada kasus, dimana fitur tidak *redundant*, maka penggunaan *information gain* sangat tepat. Tetapi pada kasus dengan banyak fitur yang *redundant*, maka harus menggunakan metode lain untuk mendeteksi korelasi antar fitur (Lee & Lee 2006).

Seleksi fitur dengan *information gain* diperkenalkan oleh Quinlan, dengan menghitung nilai entropy setiap kelas. Entropy dihitung dengan :

$$\text{Info}(D) = - \sum_{i=1}^m p_i \log_2(p_i) \quad (2.1)$$

dimana D adalah contoh data , p_i adalah bagian dari D yang memiliki kelas C_i yang dihitung dengan $\frac{|C_i D|}{D}$, dan m adalah jumlah kelas klasifikasi. Nilai maksimum dari entropy adalah 1 (teracak) dan minimum adalah 0 (terklasifikasi secara sempurna).

Tahap selanjutnya menghitung entropy fitur A dengan mempartisi nilai fiturnya terlebih dahulu. Rumus untuk menghitung informasi fitur A dengan :

$$Info_A(D) = \sum_{j=1}^v \left(\frac{|D_j|}{D} \right) \times Info(D_j) \quad (2.2)$$

Dimana D_j adalah subset dari D yang terdiri dari rentang nilai dari fitur A , dan v adalah suatu nilai yang mungkin untuk fitur A .

Information gain diukur dengan menghitung selisih antara entropy kelas dan entropy subset (entropy fitur).

$$Gain(A) = Info(D) - Info_A(D) \quad (2.3)$$

2.6. Decision Tree

Decision tree merupakan seperangkat aturan yang diterapkan untuk menghasilkan keputusan yang terklasifikasi. Algoritma C4.5 yang dibuat oleh J.R. Quinlan adalah pengembangan dari algoritma ID3 (Salzberg 1994). Boosting adalah strategi umum untuk meningkatkan hasil klasifikasi dan akurasi. Pada boosting akan memberikan bobot yang berbeda pada distribusi training data pada setiap iterasi (Pramanik et al. 2010).

2.6.1. C4.5

Algoritma C4.5 atau Classification version 4.5 merupakan pengembangan dari algoritma ID3 sehingga algoritma C4.5 mempunyai prinsip dasar kerja yang sama. Perbedaan utama C4.5 dari ID3 adalah:

- C4.5 dapat menangani fitur kontinyu dan diskrit
- C4.5 dapat menangani training data dengan missing value
- Hasil pohon keputusan C4.5 dapat dipangkas setelah dibentuk
- Pemilihan atribut yang dilakukan dengan menggunakan Gain Ratio

Algoritma C4.5 yang merupakan suksesor dari ID3 menggunakan **gain ratio** untuk memperbaiki kekurangan information gain (Quinlan 1992).

$$gain\ ratio\ (X) = \frac{gain\ (X)}{split\ info\ (X)} \quad (2.4)$$

$$split\ info\ (X) = -\sum_{i=1}^n \frac{|S_i|}{|S|} \times \log_2 \left(\frac{|S_i|}{|S|} \right) \quad (2.5)$$

dengan:

$$\frac{|S_i|}{|S|} = \text{Proporsi jumlah data pada subset } S_i$$

2.6.2. Adaboost

Boosting yang digunakan pada penelitian ini adalah adaboost. Adaboost merupakan pengembangan dari boosting. Metode ini menghasilkan serangkaian base klasifikasi. Proses latih yang digunakan untuk setiap klasifikasi berdasarkan performa klasifikasi sebelumnya.

Boosting merupakan metode yang umum digunakan untuk meningkatkan algoritma pembelajaran. Schapire pertama kali memperkenalkan algoritma boosting pada tahun 1990. Pada tahun 1995 Freund dan Schapire mengembangkannya kemudian disebut AdaBoost. Ide utama algoritma ini adalah menentukan bobot pada setiap set data latih. Pada awalnya semua bobot adalah sama, selanjutnya bobot akan diperbaharui. Jika prediksi data latih tersebut benar maka bobot akan dikurangi dan sebaliknya jika salah maka bobot klasifikasi akan ditambah (Rokach 2005). Penerapan boosting pada C4.5 memberikan akurasi yang lebih baik dari pada versi standar (C4.5) dengan menambahkan bobot pada setiap kelas data latih (Pramanik et al. 2010).

Algoritma ini mengasumsikan bahwa seperangkat data latih terdiri dari m *instances*, diberi label -1 atau 1. Klasifikasi dari *instance* dibuat dari semua klasifikasi $\{C_t\}$ yang masing-masing memiliki bobot α_t . Secara matematik dapat ditulis (Rokach 2005):

$$H(x) = \text{sign}(\sum_{t=1}^T \alpha_t \cdot C_t(x)) \quad (2.6)$$

2.7. Gwet's AC1

AC1-statistic merupakan statika untuk mengukur indeks kesepakatan antara dua pengamat. Pengukuran nilai AC1 diusulkan oleh Gwet pada tahun 2002. Gwet's AC1 menunjukkan pendekatan yang lebih dapat diandalkan dibandingkan dengan Cohen Kappa (Gwet 2002). Untuk menghitung nilai AC1, hasil observasi pengamat ditulis dalam bentuk tabel 2x2 seperti yang ditunjukkan oleh Tabel 2.1 di bawah ini:

Tabel 2.1. Tabel Penulisan Hasil Pengamatan

Pengamat 1	Pengamat 2		
	Ya	Tidak	Total
Ya	A	B	B1=A+B
Tidak	C	D	B2=C+D
Total	A1=A+C	A2=B+D	N

Pada tabel 2.1 terdapat dua pengamat yang mengklasifikasikan N subjek ke dua kemungkinan kategori. Dua kategori dilabelkan sebagai “ya” dan “tidak”. A diklasifikasikan oleh dua pengamat sebagai ya, B diklasifikasikan oleh pengamat 1 sebagai tidak dan ya oleh pengamat 2. D diklasifikasikan oleh dua pengamat sebagai tidak, dan C diklasifikasikan oleh pengamat 1 sebagai ya dan tidak oleh pengamat 2. B1 dan B2 menunjukkan jumlah subjek yang diklasifikasikan pada setiap kategori oleh pengamat 2. A1 dan A2 juga menunjukkan jumlah subjek yang diklasifikasikan pada setiap kategori oleh pengamat 1.

Perhitungan AC1-statistic :

$$\text{Kesepakatan yang terobservasi, } P = \frac{A+D}{N} \quad (2.7)$$

Dimana A adalah banyaknya data yang dikelompokkan ke dalam kategori “Ya” oleh kedua pengamat. D adalah banyaknya data yang dikelompokkan ke dalam kategori “tidak” oleh kedua pengamat.

$$\text{Probabilitas } \textit{chance-agreement}, e(y) = 2P_1(1 - P_1) \quad (2.8)$$

dimana $P_1 = \frac{(A_1+B_1)/2}{N}$ merepresentasikan perkiraan kemungkinan seorang pengamat (1 atau 2) mengelompokkan data ke dalam kategori “Ya”. A_1 dan B_1 masing-masing adalah jumlah pengamat 1 atau 2 mengelompokkan data ke dalam kategori “Ya”, sedangkan N adalah jumlah data.

$$\text{AC1-statistic, } AC1 = \frac{P-e(y)}{1-e(y)} \quad (2.9)$$

Tabel 2.3 berikut merupakan interpretasi nilai dari Gwet AC1 yang mengukur indeks kesepakatan antar dua pengamat.

Tabel 2.2. Interpretasi Nilai Gwet AC1(Landis & Koch 2012)

Index Kappa	Proporsi Kesepakatan
< 0	Rendah (<i>less than chance agreement</i>)
0.01 - 0.20	Sedikit (<i>slight agreement</i>)
0.21 - 0.40	Cukup (<i>fair agreement</i>)
0.41 - 0.60	Sedang (<i>moderate agreement</i>)
0.61 - 0.80	Substansial (<i>Substantial agreement</i>)
0.81 – 1	Hampir sempurna (<i>almost perfect</i>)

Berikut adalah contoh perhitungan AC1-static:

Terdapat dua orang pengamat yang dimintai pendapatnya terhadap 100 buah perkuliahan, apakah bermanfaat atau tidak. Hasil pengamatan terlihat seperti Tabel 2.2 di bawah ini:

Tabel 2.3. Data Pengamatan Contoh Kasus Kappa

Pengamat 1	Pengamat 2		
	Ya	Tidak	Total
Ya	40	9	49

Tidak	6	45	51
Total	46	54	100

$$P = \frac{40 + 45}{100} = 0,85$$

$$\begin{aligned} e(Y) &= 2 \left(\frac{(49 + 46)/2}{100} \right) \left(1 - \frac{(49 + 46)/2}{100} \right) \\ &= 2 \left(\frac{49 + 46}{2 \times 100} \right) \left(1 - \frac{49 + 46}{2 \times 100} \right) = 2 \times 0,475 \times 0,525 = 0,49875 \end{aligned}$$

$$AC1 = \frac{0,85 - 0,49875}{1 - 0,49875} = 0,7008$$

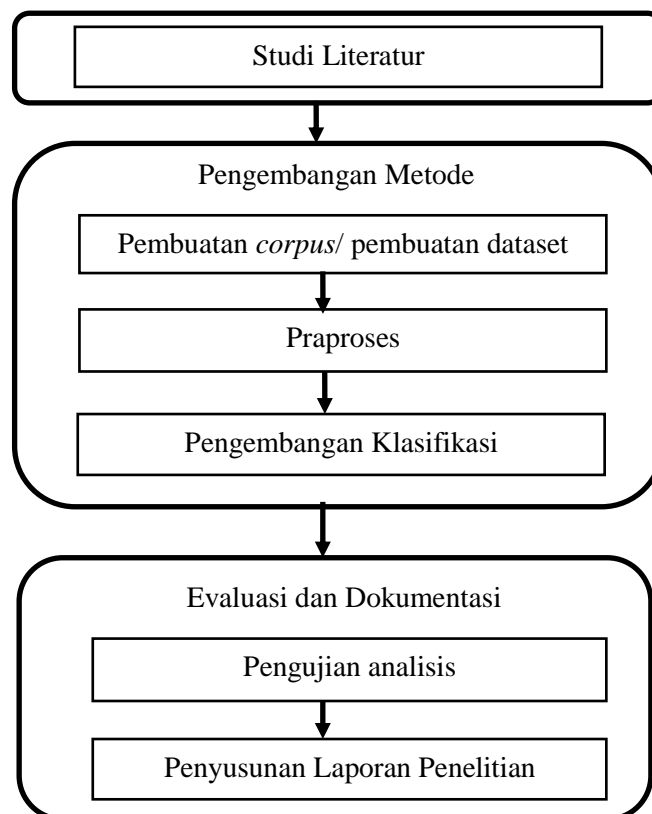
Sehingga nilai indeks kesepakatannya termasuk pada level substansial.

[Halaman ini sengaja dikosongkan]

BAB 3

METODOLOGI PENELITIAN

Untuk mencapai tujuan penelitian ini, ada beberapa langkah yang dilakukan, yaitu studi literatur, pengembangan metode pendeteksi ketidaklengkapan kebutuhan, dan evaluasi metode. Pengembangan metode pendeteksi ketidaklengkapan kebutuhan terdiri dari tiga sub-proses, yaitu pembuatan *corpus*, ekstraksi dan pemilihan fitur, dan pengembangan klasifikasi. Penulisan laporan penelitian dimulai pada awal hingga akhir penelitian. Gambar 3.1 menunjukkan langkah-langkah penelitian yang dilakukan.



Gambar 3.1 Tahapan Penelitian

3.1. Studi Literatur

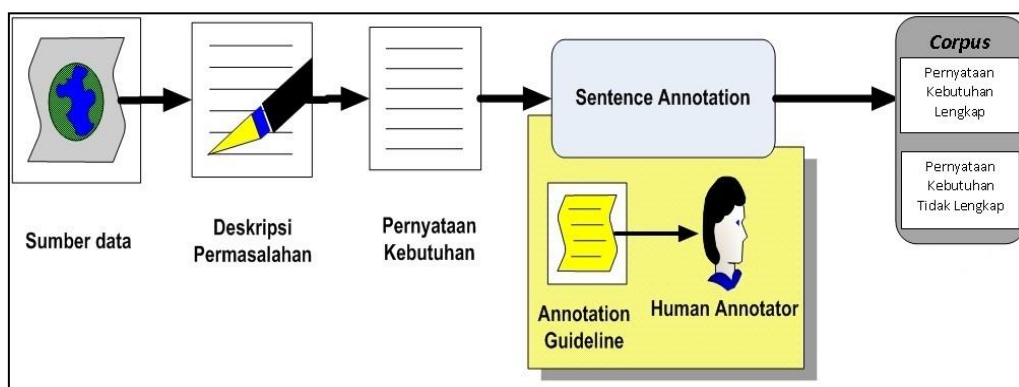
Tahap awal yang dilakukan adalah melakukan studi literature. Referensi yang digunakan pada penelitian ini adalah konferensi, jurnal, artikel dan buku yang berkaitan dengan spesifikasi kebutuhan perangkat lunak, kualitas kebutuhan perangkat lunak khususnya mengenai kelengkapan, pembuatan *corpus*, dan teknik klasifikasi menggunakan C4.5, kombinasi antara adaboost dengan C4.5.

3.2. Pengembangan Metode

Alur penelitian ini dapat dibagi menjadi 3 tahap utama yang meliputi pembuatan *corpus*, praproses dan klasifikasi.

3.2.1. Pembuatan *Corpus*

Diperlukan adanya *corpus* untuk dipergunakan untuk klasifikasi sebagai data *training* dan data *testing*. Pada penelitian ini terdapat *corpus* yang berisi kalimat-kalimat spesifikasi kebutuhan yang bersifat lengkap dan kalimat spesifikasi kebutuhan yang tidak lengkap. Akan dipilih tiga orang ahli yang menguasai dan memahami mengenai *Requirements Engineering* atau *Software Engineering* dan menguasai bahasa Inggris. Ahli akan dibekali dokumen panduan untuk melakukan anotasi, secara manual ahli akan memberikan anotasi pada setiap kalimat yang bersifat lengkap dan tidak lengkap. Gambar 3.2 mengilustrasikan proses pembuatan *corpus*.



Gambar 3.2 Pembuatan *Corpus*

Dokumen Spesifikasi Kebutuhan Perangkat Lunak yang akan dijadikan *corpus* dalam penelitian ini merupakan potongan “*problem description*” yang diambil dari *ACM’s OOPSLA DesignFest®* (<http://designfest.acm.org/>) secara acak. Dokumen ini merupakan modifikasi dataset yang digunakan oleh Ishrar Hussain(2007) dalam penelitiannya yang berjudul “*Using Text Classification to Automate Ambiguity Detection in SRS Documents*”. Dokumen menggunakan bahasa Inggris dan disimpan dalam format file text ASCII. *Problem description* berkaitan dengan spesifikasi perangkat lunak akan diekstrak menjadi pernyataan-pernyataan kebutuhan. Kemudian pernyataan kebutuhan tersebut akan dianotasi oleh ahli yang dibekali dengan dokumen panduan anotasi.

Beberapa kalimat dalam dataset tidak dilengkapi dengan tanda titik sebagai penanda akhir sebuah kalimat, oleh karena itu dilakukan normalisasi dengan menambah tanda titik di beberapa kalimat yang tidak mempunyai titik. Secara keseluruhan *corpus* terdiri dari 427 kalimat. Kemudian pernyataan kebutuhan tersebut akan dianotasi oleh ahli yang dibekali dengan dokumen panduan anotasi. *Corpus* yang telah terbentuk dipilah berdasarkan anotasinya. *Corpus_A* terdiri dari kalimat yang beranotasi tidak lengkap dan *corpus_B* terdiri dari kalimat yang beranotasi lengkap. *Corpus_A* dan *corpus_B* secara keseluruhan akan digunakan pada proses pemilihan fitur.

Dokumen panduan anotasi menjelaskan definisi pernyataan kebutuhan yang lengkap dan beberapa aturan untuk melabeli lengkap atau tidak lengkap, juga disertai dengan contoh dari masing-masing aturan. Aturan berikut terkait dengan pelabelan yang mungkin menyebabkan pernyataan kebutuhan tersebut terindikasi sebagai lengkap antara lain:

1. Menghindari penggunaan kata ganti yang merujuk pada pernyataan kebutuhan lain. Sebagai contoh :
 - a. The controller shall send the driver his itinerary for the day.
 - b. The controller shall send the driver_itinerary for the day to the Driver.Pada kalimat a merupakan contoh kalimat tidak lengkap karena terdapat kata ganti “his” yang masih memerlukan penjelasan lain siapa his tersebut.

Pada kalimat b merupakan bentuk kalimat yang lengkap dengan mengganti kata ganti *his* dengan kata benda.

2. Menghindari penggunaan kata keterangan. Sebagai contoh :

- a. The Flight Information System shall usually be on line.
- b. The Flight Information System shall have an availability of at least xx% over a period of at least yy hours.

Pada kalimat a merupakan contoh kalimat tidak lengkap karena terdapat bentuk ekspresi subjektif “*usually*” yang masih belum menjelaskan keterangan waktu secara tepat. Pada kalimat b merupakan bentuk kalimat yang lengkap dengan mengganti kata *usually* yang dijelaskan lebih rinci.

3. Menghindari penggunaan kata sifat.

- a. The Flight_Information_System shall display the Tracking_Information for relevant aircraft.
- b. The Flight_Information_System shall display the Tracking_Information of each Aircraft located less than or equal to 20 kilometers from the Airfield.

Pada kalimat a merupakan contoh kalimat tidak lengkap karena terdapat bentuk ekspresi subjektif “*relevant*” yang masih belum menjelaskan kondisi secara jelas. Pada kalimat b merupakan bentuk kalimat yang lengkap dengan mengganti kata *relevant* yang dijelaskan lebih rinci .

4. Menggunakan satuan yang jelas untuk setiap angka. Apabila tidak menggunakan maka kemungkinan tidak lengkap, sebagai contoh berikut :

- a. The Circuit_Board shall have a storage temperature or not more than 30 degrees.
- b. The Circuit_Board shall have a storage temperature or not more than 30 degrees Celsius.

Kalimat a merupakan contoh kalimat tidak lengkap karena tidak menjelaskan satuan derajat secara jelas. Dan kalimat b merupakan bentuk kalimat yang lengkap dengan memberikan satuan derajat celcius terhadap temperatur

5. Menghindari penggunaan klausa terbuka, yang memerlukan penjelasan tambahan. Seperti kata-kata *etc, and so on, not limited to*. Berikut contohnya.

a. The ATM shall display the Customer Account_Number, Account_Balance, and so on.

b. (Split into as many requirements as necessary to be complete):

The ATM shall display the Customer Account_Number.

The ATM shall display the Customer Account_Balance.

The ATM shall display the Customer Account_Overdraft_limit.

The ATM shall display the Customer

Kalimat a merupakan contoh kalimat tidak lengkap. Dan kalimat b merupakan bentuk kalimat yang lengkap dengan merinci kebutuhan secara jelas.

Untuk contoh pernyataan kebutuhan yang telah dilabeli dan merupakan hasil kesepakatan tiga orang ahli ditunjukkan pada Gambar 3.3.

Pernyataan Kebutuhan 1- Berlabel *Incomplete* :

A case may be rejected at one point and this could trigger a variant of a request, or a case lifecycle might consist of a single approval/rejection.

Pernyataan Kebutuhan 2- Berlabel *Incomplete* :

The framework should not preclude changes by the company as to the lifecycle of a case.

Pernyataan Kebutuhan 3- Berlabel *Complete* :

The system should be able to automatically create or edit a BF based on an event or a status change.

Pernyataan Kebutuhan 4- Berlabel *Complete* :

The operator must be able to plan to change every parameter for every cell.

Pernyataan Kebutuhan 5- Berlabel *Complete* :

.....

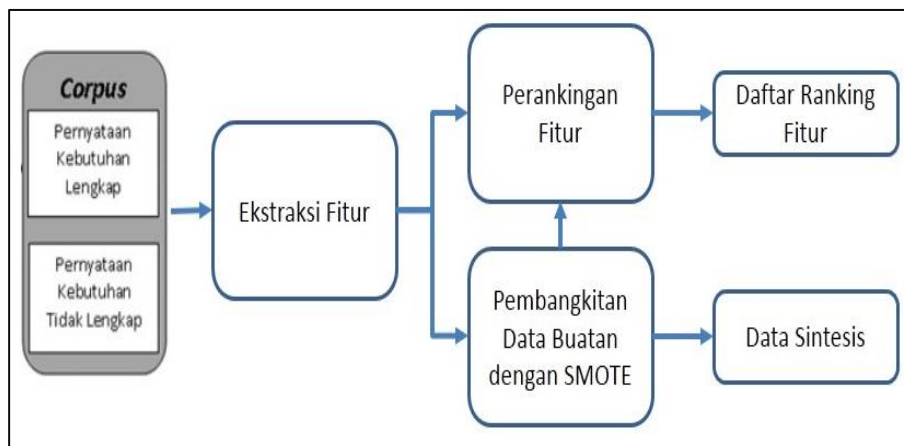
Gambar 3.3 Contoh Data Hasil *Corpus*

3.2.2. Praproses

Tahap ini bertujuan untuk membangun dataset pelatihan dan pengujian. Dataset pelatihan dan pengujian dibangun dari *corpus* yang sudah

dihasilkan dari proses sebelumnya. Pada tahap praproses terdiri dari beberapa sub proses, yaitu : proses ekstraksi fitur, pembangkitan data buatan dengan metode SMOTE, dan perankingan fitur dengan metode *information gain*.

Tahapan pada tahap praproses ini ditunjukkan pada gambar 3.4 yang menjelaskan alur kerja tahapan praproses secara umum. Seperti yang ditunjukkan pada gambar 3.4 proses preprocessing terdiri dari beberapa proses, yaitu ekstraksi fitur, pembangkitan data buatan dan perankingan fitur. Selanjutnya detail dari masing-masing proses tersebut akan dijelaskan pada masing-masing poin di subbab proses berikut.



Gambar 3.4 Praproses

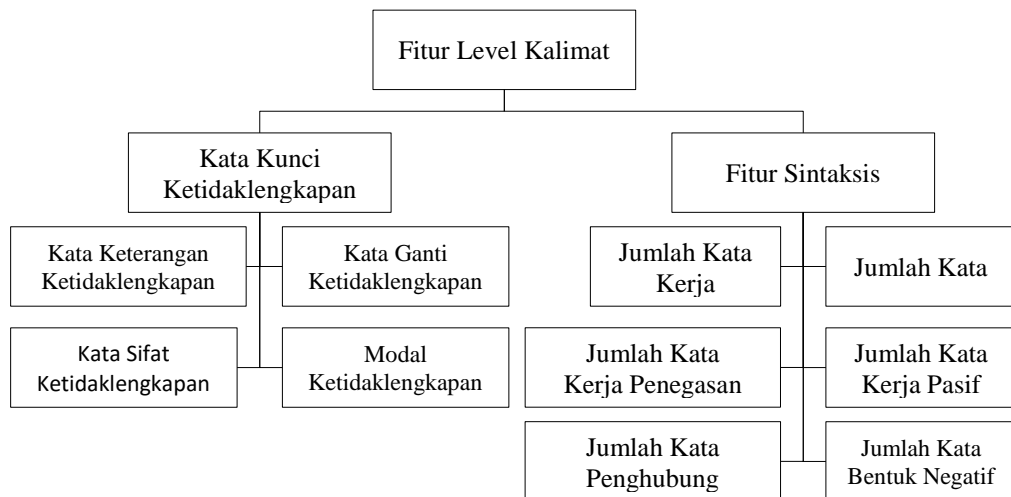
3.2.2.1. Ekstraksi Fitur

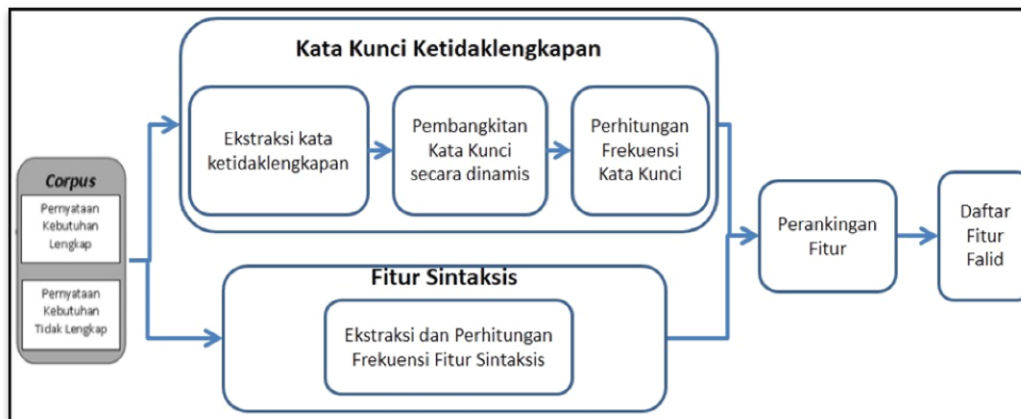
Tahap ini bertujuan untuk mengekstraksi nilai-nilai dari masing-masing fitur. Hasil dari ekstraksi fitur akan disimpan dalam format ARFF (*Attribute-Relation File Format*).

Pada tahap ini proses dimulai dengan mengidentifikasi kandidat fitur pernyataan kebutuhan yang dapat digunakan untuk mendeteksi ketidaklengkapan kebutuhan. Gambar 3.5 menunjukkan kandidat fitur pernyataan kebutuhan yang digunakan. Ekstraksi Fitur pada penelitian ini menggunakan model kualitas pernyataan kebutuhan yang diperkenalkan oleh (Genova et al. 2013) dan (INCOSE 2015) khususnya bagian untuk menilai

kualitas kebutuhan secara individu. Pendeteksian lengkap atau tidak pernyataan kebutuhan akan dilihat dari kata kunci (*keywords*) dan fitur-fitur sintaksis.

Pada penelitian ini akan mengadopsi fitur kualitas pernyataan kebutuhan yang diperkenalkan oleh (Genova et al. 2013) dan (INCOSE 2015) khususnya pada bagian kualitas kebutuhan secara individu serta menerapkan fitur-fitur tersebut pada model taxonomi yang diusulkan oleh (Hussain et al. 2007). Gambar 3.5 menunjukkan fitur yang akan dilihat dari kata kunci dan fitur-fitur sintaksis. Fitur-fitur sintaksis yang termasuk dalam level kalimat antara lain: frekuensi kata, kata kerja pasif, kata bersifat negative, jumlah kata kerja, kata kerja penegasan dan kata penghubung.





Gambar 3.5 Taxonomi Fitur

Gambar 3.6 Ekstraksi Fitur

Berdasarkan taxonomi fitur pada gambar 3.5 maka pada tahap ini akan dibagi menjadi dua tahapan, yaitu fitur kata kunci ketidaklengkapan dan fitur sintaksis. Gambar 3.6 menjelaskan alur kerja tahapan ekstraksi dan pemilihan fitur secara umum. Seperti yang ditunjukkan pada gambar 3.6 fitur kata kunci ketidaklengkapan terdiri dari tiga sub-proses, yaitu ekstraksi kata ketidaklengkapan, pembangkitan kata kunci secara dinamis, dan perhitungan frekuensi kata kunci. Sedangkan fitur sintaktis terdiri dari sub-proses yaitu ekstraksi dan perhitungan frekuensi fitur sintaktis. Setelah kedua tahapan tersebut maka akan dihasilkan nilai dari setiap fitur yang disimpan dalam format file ARFF.

1. Kata Kunci Ketidaklengkapan

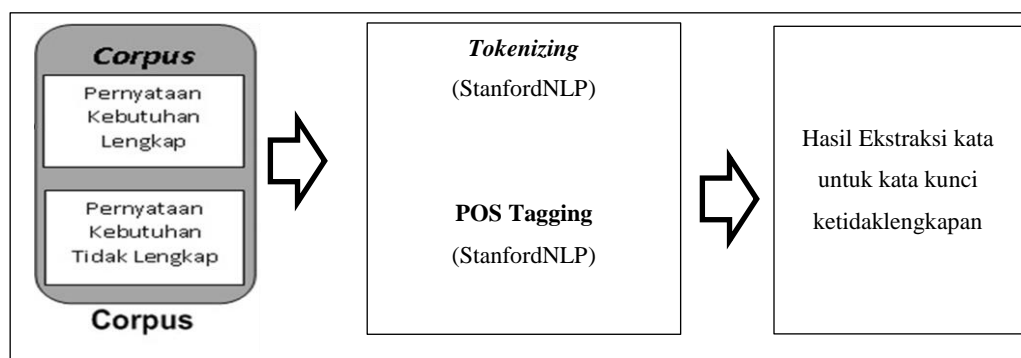
Tahap ini bertujuan untuk mengekstraksi dan membangkitkan kata kunci ketidaklengkapan secara otomatis yang hasilnya akan digunakan sebagai fitur untuk dataset pelatihan dan pengujian. Ekstraksi fitur kata kunci ketidaklengkapan terdapat tiga sub-proses, yaitu ekstraksi kata ketidaklengkapan, pembangkitan kata kunci secara dinamis, dan perhitungan frekuensi kata kunci.

a. Ekstraksi Kata

Pada penelitian ini akan menggunakan *Stanford Parser* untuk *POS tagging* dan *syntax parsing*. Fitur kata kunci ketidaklengkapan meliputi kata

keterangan, kata ganti, kata sifat, dan modal sesuai dengan taxonomi fitur pada gambar 3.5.

Tahap awal yang dilakukan adalah ekstraksi kata-kata. Setiap kalimat akan melalui proses ekstraksi kata untuk mendapatkan *tag* setiap kata yang ada pada kalimat tersebut. Gambar 3.7 di bawah ini menunjukkan alur dalam ekstraksi kata.



Gambar 3.7 Tahapan Eksraksi Kata untuk Kata Kunci Ketidاكلengkapan

Seperti yang dijelaskan pada gambar 3.7, pada tahap ini terdapat beberapa sub proses yang antara lain sebagai berikut :

- *Tokenizing*

Tokenizing merupakan proses memecah dokumen maupun kalimat menjadi kata atau *term*. Proses *tokenizing* dalam penelitian ini menggunakan *library Stanford Natural Language Processing*. Sebagai contoh pernyataan kebutuhan 1 pada Gambar 3.3 apabila dilakukan proses *tokenizing* akan menjadi sebagai berikut.

a(1), case(2), may(3), be(4), rejected(5), at(6), one(7), point(8), and(9), this(10), could(11), trigger(12), a(13), variant(14), of(15), a(16), request(17), or(18), a(19), case(20), lifecycle(21), might(22), consist(23), of(24), a(25), single(26), approval(27), rejection(28)

Pada proses ini dihasilkan 28 kata yang diperoleh dari pernyataan kebutuhan 1.

- POS Tagging

POS Tagging merupakan proses dilakukan untuk mendapatkan informasi *tag* dari setiap kata yang ada pada dokumen. Proses ini memanfaatkan *library* POS tagger dari Stanford yang sudah ada dan dapat diunduh secara gratis di situs resmi NLP Stanford.

Sebagai contoh pernyataan kebutuhan 1 pada Gambar 3.3 apabila dilakukan proses POS Tagging akan menjadi seperti berikut:

[a_DT,case_NN,may_MD,be_VB,rejected_VBN,at_IN,one_CD,point_NN, and_CC,this_DT,could_MD,trigger_VB,a_DT,variant_NN,of_IN,a_DT,request_NN,or_CC,a_DT,case_NN,lifecycle_NN,might_MD,consist_VB,of_IN,a_DT,single_JJ,approval_NN,rejection_NN]

Kemudian setiap kata dari *corpus_A* (tidak lengkap) tersebut diidentifikasi apakah merupakan kategori POS(kata keterangan, kata sifat, kata ganti, dan modal), selanjutnya disimpan di tabel frekuensi untuk dijadikan sebagai kandidat kata kunci. Berikut ini tag yang dipergunakan untuk mengidentifikasi masing-masing kategori:

a. Kata ganti

Kata ganti ditag dengan “PRP” atau “PRP\$”.

b. Kata keterangan

Kata keterangan ditag dengan “RB” untuk kata keterangan, “RBR” untuk kata keterangan *comperative* dan “RBS” untuk kata keterangan *superlative*.

c. Kata Modal

Kata yang ditandai dengan “MD” maka sebaiknya dihindari karena mengandung ekspresi kondisional, kecuali kata *Must*.

d. Kata sifat

Kata sifat ditag dengan “JJ” untuk kata sifat, “JJR” untuk kata sifat yang *comperative* dan “JJS” untuk kata sifat yang *superlative*.

Dari contoh pada gambar 3.3, maka kandidat kata kunci dari masing-masing kategori POS (kata ganti, kata keterangan, modal, dan kata sifat) ditunjukkan pada Tabel 3.1:

Tabel 3.1 Contoh Kandidat Kata Kunci dari Masing-masing Kategori

Kategori	Kata
Kata sifat (JJ)	<i>Single</i>
Kata keterangan (RB)	<i>Not</i>
MD	May
	could
	Might
	Should
Kata ganti (PRP)	-

b. Pembangkitan Kata Kunci secara Dinamis

Tahap pembangkitan kata kunci bertujuan membentuk kamus kata berisi daftar kata yang mungkin mengindikasikan sebagai ketidaklengkapan. Untuk membuat kamus yang berisi ketidaklengkapan kata kunci maka tahapannya :

- Setelah melakukan ekstraksi kata pada setiap kalimat.
- Menghitung rasio frekuensi kemunculan kata pada *corpus* tidak lengkap dan rasio kemunculan pada semua *corpus* untuk untuk setiap kategori POS(kata sifat, kata keterangan, kata ganti, dan kata *modal*) secara terpisah. Untuk setiap kata kunci rasio ini disebut “*Likelihood Ratio*” (LR).

$$LR(\text{kata } x) = \frac{\text{freq}(\text{kata } x) \text{ di } \text{corpus}_A}{\text{freq}(\text{kata } x) \text{ di } \text{corpus}_A + \text{freq}(\text{kata } x) \text{ di } \text{corpus}_B} \quad (3.1)$$

- Selanjutnya menghitung nilai baseline LR sebagai nilai *cut-off threshold* untuk setiap kategori POS.

$$\text{Baseline LR} = \frac{\text{Jumlah kata di } \text{corpus}_A}{(\text{Jumlah kata di } \text{corpus}_A + \text{Jumlah kata di } \text{corpus}_B)} \quad (3.2)$$

- Membandingkan nilai LR dengan baseline LR dari setiap kata.
- Kata yang memiliki nilai LR lebih tinggi dari baseline LR akan disimpan sebagai kata kunci yang mengindikasikan ketidaklengkapan, dan disimpan pada list *bad* sesuai dengan nama POSnya (misal *bad_MD*).

Berdasarkan contoh dari Gambar 3.3 dan Tabel 3.1 digunakan kategori MD untuk contoh tahap selanjutnya yang ditunjukkan pada Tabel 3.2. Tabel 3.2 menunjukkan contoh hasil pembangkitan kata kunci yang lebih dari LR pada sejumlah kata kunci bertipe modal. Dari table 3.2 terlihat bahwa kata *may*,

could, dan *might* memiliki nilai LR lebih besar dari baseline LR maka akan disimpan sebagai kata kunci dengan list *bad_MD*.

Tabel 3.2 Contoh LR dari kata kunci berkategori POS MD

Kata Kunci	POS/Kategori	Frekuensi di corpus _A	Frekuensi di corpus _B	LR
may	MD	1	0	1
could	MD	1	0	1
might	MD	1	0	1
<i>Baseline LR = 0.55</i>				
<i>Should</i>	MD	1	1	0.5

c. Perhitungan Frekuensi Kata Kunci

Langkah selanjutnya menghitung frekuensi kata kunci ketidaklengkapan sesuai dengan kata kunci yang tersimpan di tahap sebelumnya pada setiap kalimat pernyataan kebutuhan. Kata yang sesuai dengan kata kunci ketidaklengkapan akan dihitung dan menyimpannya ke dalam dengan format khusus yang nantinya bersama dengan fitur sintaksis akan digunakan sebagai data latih dan data uji.

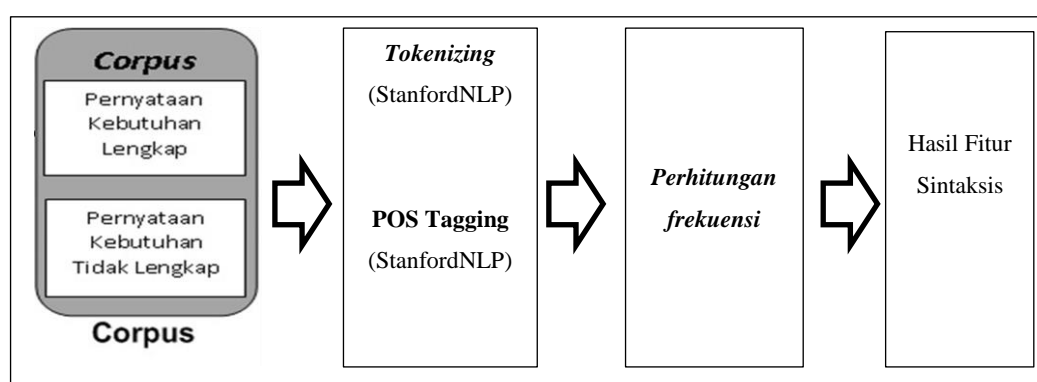
Dari contoh pada Gambar 3.3 maka dihitung frekuensi fitur kata kunci yang ada pada setiap pernyataan kebutuhan seperti Tabel 3.3 berikut :

Tabel 3.3 Contoh Hasil Perhitungan Frekuensi Fitur Kata Kunci

Pernyataan Kebutuhan ke-	Kata sifat (bad_jj)	Kata Keterangan (bad_rb)	Kata Ganti (bad_prp)	Modal (bad_md)
1	1	0	0	3
2	0	1	0	1
3	0	0	0	1
4	0	0	0	0
5	0	0	0	0

2. Fitur Sintaktis

Tahap ini bertujuan untuk mengekstraksi fitur sintaktis yang hasilnya bersama-sama dengan hasil fitur kata kunci akan digunakan sebagai fitur untuk dataset pelatihan dan pengujian. Sama seperti pada proses fitur kata kunci ketidaklengkapan, proses ini juga memanfaatkan *Stanford Parser Brill's POS* dan kamus kata sentiwordnet. *Stanford Parser Brill's POS* dilengkapi *tagger* dan *dependencies*, sedangkan sentiwordnet dilengkapi daftar kata-kata yang memiliki *sense* dengan nilai positif dan negatif. Langkah ini meliputi ekstraksi kata dan perhitungan frekuensi fitur sintaktis.



Gambar 3.8 Tahapan Eksraksi Kata dan Perhitungan Frekuensi Fitur Sintaksis

Ekstraksi dan Perhitungan Frekuensi Fitur Sintaksis

Sesuai dengan taxonomi fitur pada gambar 3.8 fitur sintaksis meliputi jumlah kata, jumlah kata kerja, jumlah kata kerja pasif, jumlah kata menunjukkan penegasan, jumlah kata penghubung, dan kata bentuk negatif.

Pada gambar 3.8 menjelaskan tahapan ekstraksi kata serta perhitungan frekuensi untuk fitur sintaksis. Langkah pertama dilakukan adalah ekstraksi kata-kata. Sama seperti tahapan ekstraksi kata pada kata kunci ketidaklengkapan, ekstraksi kata meliputi sub proses *tokenizing* dan POS tagging pada setiap kalimatnya. Sebagai contoh pernyataan kebutuhan 2 pada Gambar 3.3 berikut hasil proses *tokenizing* :

the(1), framework(2), should(3), not(4), preclude(5), changes(6), by(7), the(8), company(9), as(10), to(11), the(12), lifecycle(13), of(14), a(15), case(16)

Pada proses ini dihasilkan 16 kata yang diperoleh dari pernyataan kebutuhan 2.
Hasil POS tagger nya :

*the_DT,framework_NN,should_MD,not_RB,preclude_VB,changes_NNS,
by_IN,the_DT,company_NN,as_IN,to_TO,the_DT,lifecycle_NN,of_IN,
a_DT,case_NN*

Setelah dilakukan proses *tokenizing* dan POS tagging ekstraksi kata maka dihitung berapa jumlah dari masing-masing fitur sintaksis pada kalimat tersebut. Berikut ini tag dan fitur yang termasuk kategori fitur sintaksis.

a. Jumlah kata

Dilakukan pemenggalan kata sehingga setiap kata dapat berdiri sendiri.
Selanjutnya menghitung jumlah kata dalam kalimat tersebut.

b. Jumlah kata kerja

Menghitung kata yang ditandai dengan VB, VBD, VBG, VBN, VBZ dan VBP, yang masing-masing menunjukkan sebagai kata kerja.

c. Jumlah kata kerja pasif

Menggunakan *Stanford Dependencies*, jika terdapat *auxpass* maka terdapat kata kerja termasuk kata pasif.

d. Jumlah kata menunjukkan penegasan

Dilakukan pemenggalan kata sehingga setiap kata dapat berdiri sendiri.
Selanjutnya mengidentifikasi kata yang terdiri dari *must*, *have to*, *has to*, dan *had to*.

e. Jumlah kata penghubung

Kata penghubung ditag dengan inisial "CC".

f. Jumlah kata bentuk negatif

Untuk fitur jumlah kata bentuk negative selain menggunakan library Stanford juga menggunakan kamus kata SentiWordNet. SentiWordNet digunakan untuk mengidentifikasi kata tersebut negatif atau tidak. Setiap kata pada SentiWordNet memiliki *sense* yang berbeda dengan nilai positif dan negatif yang berbeda juga. Untuk menentukan nilai positif atau negatif

suatu kata maka pada penelitian ini menggunakan metode interpretasi SentiWordNet.

Untuk mendapatkan nilai interpretasi SentiWordNet dilakukan dengan menghitung rata-rata nilai positif dan negative dari setiap kata berdasarkan POS yang dimiliki pada SentiWordNet. Pada penelitian ini hanya berfokus pada kata yang memiliki nilai negatif, maka pada penelitian ini hanya mencari nilai interpretasi negatifnya saja.

Setelah proses POS tagging dari setiap kata. Setiap kata akan dicari nilai interpretasi negatif SentiWordNetnya. Karena terdapat perbedaan penamaan tag pada POS Tagger NLP Stanford dengan SentiWordnet maka sebelum kata dicocokkan dengan basis data SentiWordNet harus dilakukan penyamaan tagging terlebih dahulu. Tabel 3.4 menjelaskan tentang penyamaan tagging antara POS Tagger NLP dengan SentiWordNet.

Tabel 3.4 Penyamaan Tag dengan SentiWordNet

No	POS Tagger	SentiWordNet
1	<i>NN (Noun, singular or mass)</i>	<i>n (noun)</i>
2	<i>NNS (Noun, plural)</i>	
3	<i>NNP (Proper noun, singular)</i>	
4	<i>NNPS (Proper noun, plural)</i>	
5	<i>JJ (Adjective)</i>	<i>a (adjective)</i>
6	<i>JJR (Adjective, Comparative)</i>	
7	<i>JJS (Adjective, Superlative)</i>	
8	<i>RB (Adverb)</i>	<i>r (adverb)</i>
9	<i>RBR (Adverb, Comparative)</i>	
10	<i>RBS (Adverb, Superlative)</i>	
11	<i>VB (Verb, Base Form)</i>	<i>v (verb)</i>
12	<i>VBD (Verb, Past Tense)</i>	
13	<i>VBG (Verb, Gerund or Present Participle)</i>	
14	<i>VCN (Verb, Past Participle)</i>	
15	<i>VBP (Verb, non-3rd person singular person)</i>	
16	<i>VBZ (Verb, 3rd person singular person)</i>	

Setelah proses penyamaan dan pencocokan dengan basis data SentiWordNet selesai, maka langkah selanjutnya menghitung nilai interpretasi negatif dari setiap kata sesuai dengan POS. Interpretasi negatif diperoleh dengan merata-rata nilai negatif dari suatu kata yang sama pos nya

Setelah mendapat nilai interpretasi negatif SentiWordNet, maka memberikan nilai *threshold*(ambang batas) sebesar 0.5. Pemberian nilai ambang batas untuk menyeleksi kata-kata yang dinilai bersifat negative. Penentuan ambang batas tersebut berdasarkan hasil percobaan dengan memperhatikan hasil nilai interpretasi dari setiap kata. Jika nilai interpretasi lebih dari sama dengan 0.5 maka kata tersebut dinilai sebagai term negatif. Berikut merupakan contoh dari Gambar 3.3 pernyataan kebutuhan 2.

Hasil POS tagger nya :

*the_DT,framework_NN,should_MD,not_RB,preclude_VB,changes_NNS,
by_IN,the_DT,company_NN,as_IN,to_TO,the_DT,lifecycle_NN,of_IN,
a_DT,case_NN*

Tabel 3.5 Perhitungan Nilai Interpretasi Negatif SentiWordNet

<i>Term</i>	POS Tag	Tag (SentiWordNet)	Nilai Interpretasi Negatif
<i>Framework</i>	NN	N	0.0
<i>Not</i>	RB	R	0.625
<i>Preclude</i>	VB	V	0.0
<i>Changes</i>	NNS	N	0.0
<i>Company</i>	NN	N	0.0
<i>Lifecycle</i>	NN	N	0.0
<i>Case</i>	NN	N	0.0

Dari tabel 3.5 hanya satu kata yaitu *not* yang memiliki nilai interpretasi negative lebih dari 0.5, maka jumlah kata negative dalam sebuah kalimat ada 1.

Dari contoh pada Gambar 3.3 maka dihitung frekuensi fitur sintaksis yang ada pada setiap pernyataan kebutuhan seperti Tabel 3.6 berikut :

Tabel 3.6 Contoh Hasil Perhitungan Frekuensi Fitur Sintaksi

Pernyataan Kebutuhan ke-	Jumlah Kata	Jumlah Kata Kerja	Jumlah Kata Kerja Pasif	Jumlah Kata Penegasan	Jumlah Kata Penghubung	Jumlah Kata Negatif
1	28	4	1	0	2	1
2	16	1	0	0	0	0
3	20	4	0	1	2	0
4	14	3	0	0	0	0
5	11	3	0	0	1	0

Semua hasil perhitungan frekuensi fitur sintaksis dan fitur kata kunci akan disimpan ke dalam dengan format khusus yang nantinya akan digunakan sebagai data latih dan data uji. Dari contoh pada Gambar 3.3, maka dihasilkan ekstraksi fitur seperti Tabel 3.7 yang merupakan gabungan dari fitur sintaksis dan fitur kata kunci.

Tabel 3.7 Contoh Hasil Ekstraksi Fitur

Kebutuhan ke-	Kt sifat (bad_jj)	Kt Keterangan (bad_rb)	Kt Ganti (bad_prp)	Mo dal (bad_md)	Jml Kata	Jml Kata Kerja	Jml Kata Kerja	Jml Kata Penegasan	Jml Kata Penghubu	Jml Kata Negatif	Label
1	1	0	0	3	28	4	1	0	2	1	Incomplete
2	0	1	0	1	16	1	0	0	0	0	Incomplete
3	0	0	0	1	20	4	0	1	2	0	Complete
4	0	0	0	0	14	3	0	0	0	0	Complete
5	0	0	0	0	11	3	0	0	1	0	Complete

Karena *classifier* akan dibuat menggunakan Weka, maka *Training file* akan disesuaikan dengan format Weka yaitu *Attribute-Relation File Format* (ARFF).

3.2.2.2. Pembangkitan Data Buatan dengan SMOTE

Tahap pembangkitan data buatan dengan algoritma SMOTE(*Synthetic Minority Oversampling Technique*) berfungsi untuk memperbanyak data dari kelas yang minoritas agar sebanding dengan kelas mayoritas. Pada proses pembuatan corpus menghasilkan data berlabel lengkap lebih banyak dari pada data yang berlabel tidak lengkap. Pada penelitian ini kelas minor yaitu tidak lengkap dibangkitkan data buatan sehingga jumlah data dari kelas minor sama atau sebanding dengan data kelas mayor yaitu kelas lengkap.

Tahap pembangkitan data buatan dimulai dari hasil ekstraksi fitur yang berupa file ARFF. Gambar 3.9 merupakan contoh beberapa data dari kelas minoritas *incomplete*.

BAD_JJ Numeric	BAD_RB Numeric	BAD_PRP Numeric	BAD_MD Numeric	jml_KATA Numeric	jml_KT_KERJA Numeric	jml_KT_PENGHUBUNG Numeric	jml_KT_PENEGASAN Numeric	jml_KT_PASIF Numeric	jml_KT_NEGATIF Numeric	Class Nominal
2.0	0.0	0.0	0.0	10.0	1.0	1.0	1.0	0.0	0.0	Incomplete
1.0	0.0	0.0	1.0	23.0	4.0	0.0	0.0	1.0	0.0	Incomplete
1.0	0.0	0.0	1.0	14.0	3.0	0.0	0.0	0.0	0.0	Incomplete
0.0	0.0	0.0	0.0	16.0	2.0	1.0	0.0	0.0	0.0	Incomplete
1.0	0.0	0.0	1.0	27.0	4.0	1.0	0.0	1.0	0.0	Incomplete

Gambar 3.9 Contoh Data ARFF

Langkah pertama adalah menghitung jumlah kelas minoritas pada penelitian ini adalah kelas *incomplete* yang sebanyak 141. Selanjutnya menentukan jumlah data kelas buatan yang akan dihasilkan, pada penelitian ini dibuat 100% dari kelas *incomplete* yang berarti sebanyak dua kali lipat dari data kelas *incomplete* sebelumnya.

Selanjutnya memilih secara random tetangga terdekat(\ddot{x}) dari elemen kelas *incomplete* untuk setiap masing-masing fitur. Pada contoh ini dihitung untuk fitur kata sifat ketidaklengkapan (*bad_jj*) dengan tetangga terdekat(\ddot{x}) adalah 1. Untuk salah satu *randomseed*(x_i) adalah 2 dari fitur *bad_jj*. Maka langkah selanjutnya menghitung perbedaan antara (\ddot{x}) dengan (x_i) seperti berikut :

$$(\ddot{x} - x_i) = |1 - 2| = |-1|$$

Selanjutnya mengalikan $(\bar{x} - x_i)$ dengan angka random antara 0 sampai 1. Dan menambahkan dengan x_i dan nilai random 0.304, seperti berikut :

$$x_{baru} = x_i + (\bar{x} - x_i) \times rand(0 - 1)$$

$$x_{baru} = 2 + |-1| \times 0.304 = 2.304$$

Maka data buatan yang dihasilkan oleh metode SMOTE berupa angka pecahan, hal ini harus dirubah terlebih dahulu menjadi bentuk numerik dengan membulatkan angka dibelakang koma.

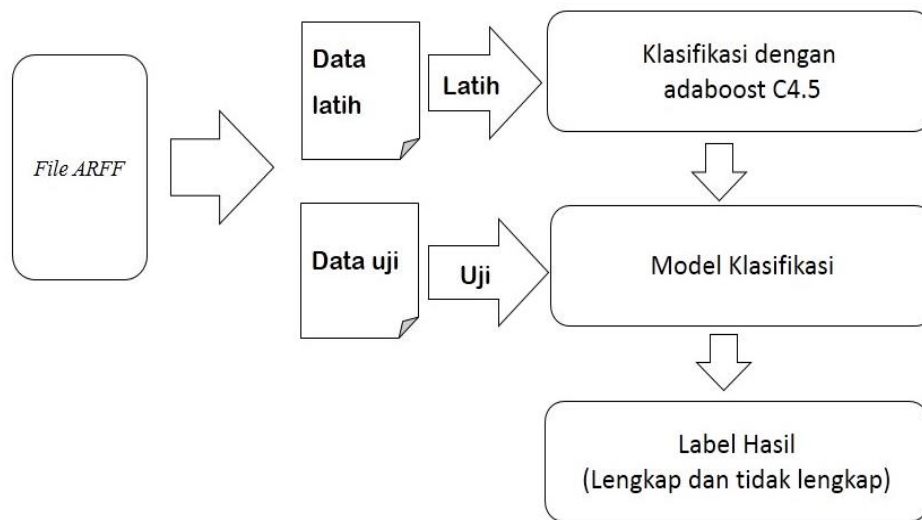
3.2.2.3. Perangkingan Fitur

Perangkingan fitur digunakan untuk memilih fitur yang valid dan fitur yang berpengaruh pada hasil klasifikasi dengan melakukan beberapa kali ujicoba pada klasifikasi. Implementasi algoritma seleksi fitur *information gain* dijalankan dengan memanfaatkan library Weka. Tahapan dalam perangkingan fitur sebagai berikut:

- a. Menghitung nilai entropy kelas dengan persamaan 2.1
- b. Mengurutkan nilai dari suatu fitur dan mempartisinya untuk dihitung entropy fitur dengan persamaan 2.2
- c. Menghitung nilai gain dengan persamaan 2.3.
- d. Memilih nilai gain tertinggi.
- e. Melakukan tahap a sampai d untuk setiap fitur.
- f. Meranking nilai gain dari masing-masing fitur.

3.2.3. Pengembangan Klasifikasi

Tahap ini bertujuan untuk membangun klasifikasi yang menggunakan kombinasi *adaboost* dengan C4.5. Klasifikasi akan mempunyai modul latih yang akan menghasilkan aturan model klasifikasi secara dinamis berdasarkan data dari *corpus* yang telah dibuat. Model klasifikasi akan mengklasifikasi sebuah kalimat apakah termasuk lengkap atau tidak.



Gambar 3.10 Tahapan Klasifikasi

Gambar 3.10 merupakan gambaran tahapan klasifikasi. Setelah tahapan praproses maka menghasilkan data yang berformat ARFF. Data selanjutnya dibedakan menjadi data latih dan data uji. Data latih dilakukan klasifikasi terlebih dahulu untuk membentuk model klasifikasi. Selanjutnya data uji akan dilakukan klasifikasi sesuai hasil model dari proses latih sebelumnya. Klasifikasi akan dibangun memanfaatkan perangkat lunak *Weka* yang menyediakan implementasi kombinasi antara adaboost dengan C4.5.

3.3. Analisis Pengujian

Tahap pengujian bertujuan untuk membuktikan model klasifikasi yang dibangun dengan fitur-fitur yang diusulkan dapat diandalkan. Pada penelitian ini akan digunakan *Gwet's AC1* untuk menilai hasil klasifikasi yang telah dibuat. Nilai AC1 akan menjadi tolok ukur apakah kerangka kerja yang diusulkan dapat diandalkan untuk mendeteksi adanya ketidaklengkapan pada pernyataan kebutuhan perangkat lunak. Model klasifikasi dibangun dengan menggunakan *corpus_A* yang berisi kalimat tidak lengkap dan *corpus_B* berisi kalimat lengkap dari dua orang ahli, masing-masing corpus dari tiga orang ahli tersebut akan diujicobakan dan dihitung nilai *Gwet's AC1*.

Pengujian penelitian ini menerapkan teknik crossvalidasi sebesar 10 fold. Data dibagi menjadi 10 bagian yang diambil secara acak. Selanjutnya

dilakukan perulangan dengan sembilan dari sepuluh bagian akan dijadikan sebagai latih dan sisanya akan menjadi uji. Begitu juga untuk perulangan kedua sampai setiap bagian pernah menjadi data uji dan data latih.

Pengujian dengan teknik crossvalidasi ini diterapkan kepada kedua data (data asli dan data SMOTE). Selanjutnya membandingkan hasil Gwet AC1 dari kedua data tersebut. Selain teknik crossvalidation, untuk keperluan analisa pada penelitian ini juga melakukan pembagian data menjadi *training* dan *testing* untuk mengetahui data-data mana saja yang salah prediksi atau berbeda dari pakar dengan kerangka kerja.

BAB 4

UJI COBA DAN EVALUASI

Pada bab ini akan dijelaskan mengenai uji coba dan evaluasi penelitian yang telah dilakukan. Pada bab ini akan dibagi menjadi tiga bagian yaitu implementasi penelitian, uji coba, dan analisis uji coba. Lingkungan uji coba meliputi perangkat keras dan perangkat lunak yang digunakan untuk implementasi system.

4.1. Implementasi Penelitian

Dalam penelitian ini sistem untuk implemetasi dan ujicoba akan dibangun dalam lingkungan pengembangan sebagai berikut :

Sistem Operasi	: Windows 7 64 bit
RAM	: 2 GB
Processor	: Intel Core i5
IDE	: Netbiens 8.02
DBMS	: MySQL
Software Pendukung	: XAMPP v3.2.2, Google Chrome
Library	: Stanford corenlp 3.6.0
Weka	: weka 3.7.3

4.2. Perancangan Uji Coba

Pada sub bab ini menjelaskan tentang perancangan ujicoba yang dilakukan dalam penelitian ini. Perancangan ujicoba tersebut dibagi menjadi beberapa tahap yang antara lain adalah pembentukan *corpus* dan pembagian dataset, scenario ujicoba, dan implementasi ujicoba.

4.2.1. Pembentukan Corpus

Dataset yang digunakan pada penelitian ini adalah dataset hasil pembentukan *corpus* oleh beberapa orang ahli. Dua orang ahli merupakan dosen yang mengajar matakuliah Rekayasa Perangkat Lunak dan menguasai

bahasa Inggris. Orang ahli lainnya pernah meneliti tentang Rekayasa Kebutuhan dan menguasai bahasa Inggris.

Jumlah keseluruhan data adalah sebanyak 427 pernyataan kebutuhan, setiap pernyataan kebutuhan masing-masing akan dilabeli oleh tiga orang ahli. Sehingga apabila ketiga ahli sepakat tidak lengkap maka pernyataan kebutuhan tersebut berlabel tidak lengkap, dan bila ketiganya sepakat lengkap maka berlabel lengkap. Namun, bila terdapat ahli yang berbeda pendapat maka diambil suara terbanyak dari dua orang ahli yang sependapat.

Dari 427 pernyataan kebutuhan pada dataset terdapat 133 pernyataan menurut ketiga ahli terindikasi lengkap dan 48 pernyataan menurut ketiga ahli terindikasi sebagai tidak lengkap. Sedangkan sisanya menurut dua dari tiga orang ahli sepakat sebanyak 153 pernyataan terindikasi lengkap dan 93 pernyataan terindikasi lengkap. Sehingga hasil pembentukan *corpus* sebanyak 286 pernyataan kebutuhan berlabel lengkap dan 141 pernyataan kebutuhan berlabel tidak lengkap.

Selanjutnya *corpus* yang terbentuk akan digunakan untuk proses selanjutnya yaitu pembangkitan kata kunci dan ekstraksi fitur sampai menghasilkan dokumen ARFF.

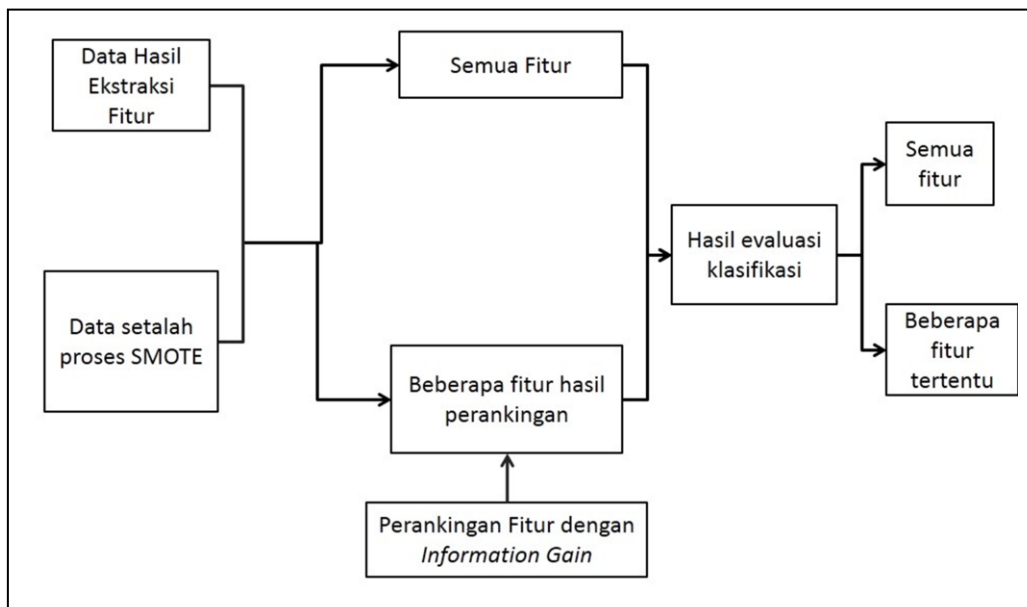
4.2.2. Skenario Uji Coba

Berdasarkan metode yang diusulkan, penelitian memiliki beberapa tahapan proses dalam menentukan hasil pendeteksian ketidaklengkapan kebutuhan. Tahapan tersebut sudah dijelaskan bab sebelumnya. Ujicoba dalam penelitian ini menggunakan dua macam data yaitu, data hasil ekstraksi fitur sebanyak 427 dan data hasil ekstraksi fitur setelah proses SMOTE sebanyak 568.

Data setelah proses SMOTE merupakan data asli yang berjumlah 427 ditambahkan dengan data buatan hasil SMOTE sebanyak 141. Terdapat dua model pengujian yaitu pengujian dengan semua fitur dan pengujian dengan beberapa fitur hasil perankingan dari *information gain*.

Dengan kata lain dalam penelitian ini ada 2 hasil yang dikeluarkan oleh setiap macam data yaitu hasil kappa dengan Gwet AC1-statistic dari

klasifikasi dengan semua fitur dan hasil klasifikasi dengan beberapa fitur. Gambar 4.1 dibawah ini menjelaskan tentang rencana pengujian dalam penelitian ini.



Gambar 4.1 Rencana Ujicoba

Berdasarkan Gambar 4.1, proses pengujian yang dilakukan dengan menggunakan dua macam data yaitu data hasil ekstraksi fitur dan data hasil ekstraksi fitur setelah proses SMOTE. Kedua macam data tersebut masing-masing diujikan pada dua model pengujian yaitu pengujian dengan menggunakan semua fitur dan beberapa fitur tertentu dimana fitur yang memiliki ranking terendah sesuai hasil metode *information gain* tidak akan digunakan. Hasil dari kedua pengujian tersebut nantinya akan dianalisa sebagai bahan evaluasi dalam penelitian ini.

4.2.3. Implementasi Uji Coba

Pada sub bab ini dijelaskan tentang detail implementasi ujicoba yang dilakukan dalam penelitian ini. Implementasi pengujian tersebut meliputi hasil pembangkitan kata kunci ketidaklengkapan, pembentukan ARFF, perankingan

fitur dengan *information gain*, pengujian dengan semua fitur dan pengujian beberapa fitur tertentu.

4.2.3.1. Pembangkitan Kata Kunci Ketidaklengkapan

Hasil pembangkitan kata kunci ketidaklengkapan secara otomatis bertujuan membentuk daftar kata-kata yang terindikasi sebagai kata yang menunjukkan ketidaklengkapan. Hasil pembangkitan kata kunci secara otomatis dapat dilihat pada tabel 4.1. Dari table 4.1 terlihat bahwa kata kunci dikelompokkan sesuai jenisnya, yaitu bad_JJ(kata sifat ketidaklengkapan), bad_RB(kata keterangan ketidaklengkapan), bad_PRP(kata ganti ketidaklengkapan), dan bad_MD(modal ketidaklengkapan).

Tabel 4.1 Tabel Hasil Pembangkitan Kata Kunci

Kelompok Kata	Daftar Kata Kunci Ketidaklengkapan
bad_JJ	<i>equivalent,accurate,bad,desirable,inactive,real-time, on-line, above, similar, large, old, lower, personal, optional, easy, lowest, slow,appropriate,accessible, internal, other, alive, least>manual, current, same, overdue, established, outstanding, necessary, efficient, downstream, possible,token,single,connected,such, fast,able, certain, first, essential, scalable, small, acceptable, compact, adequate, own, several, simple, good, centralized, different, new, individual, particular, less, multi-byte, read-out, most, formal, safe, final, fixed, basic,present, largest, inexpensive, financial, proper, random, tunable, total, public, parallel, responsible, original, application-specific, multiple, clear, enough, specific, intelligent, third, detailed, corresponding, due, various, groupable, multi-lingual,online,compliant,chronological,base</i>
bad_RB	<i>very,quickly, back, recursively, periodically, immediately, elsewhere, intricately, improperly, precisely, not, later, easily,usually, so, gracefully, directly, just, fully, otherwise, successfully, unfrequently, recently, also, often, then, statistically, dynamically, as, currently, strongly,efficiently, relatively,significantly</i>
bad_PRP	<i>his,themselves,its,them,it,he,they</i>
bad_MD	<i>can,may,will,could,might,should</i>

4.2.3.2. Pembentukan Attribute-Relation File Format (ARFF)

Tahap ekstraksi fitur mengekstrak nilai fitur untuk setiap kalimat di *corpus_A* dan *corpus_B* dan menyimpannya untuk digunakan sebagai data latih dan data uji. Pada penelitian ini menggunakan Weka untuk membangun model klasifikasi, maka format untuk data agar bisa dibaca oleh Weka harus *Attribute-Relation File Format (ARFF)*.

Bagian dari file bertipe ARFF ditunjukkan pada gambar 4.2. Bagian header file terdiri dari deklarasi @ATTRIBUTE dari setiap fitur. Terdapat deklarasi @ATTRIBUTE yang disebut *class* terdiri dari “*incomplete*” dan “*complete*”. Setiap baris di bawah deklarasi @DATA merupakan nilai dari masing-masing fitur dan kelas dari hasil anotasi yaitu *incomplete* atau *complete*.

```
@RELATION sentence_corpus

@ATTRIBUTE BAD_JJ NUMERIC
@ATTRIBUTE BAD_RB NUMERIC
@ATTRIBUTE BAD_PRP NUMERIC
@ATTRIBUTE BAD_MD NUMERIC
@ATTRIBUTE jml_KATA NUMERIC
@ATTRIBUTE jml_KT_KERJA NUMERIC
@ATTRIBUTE jml_KT_PENGHUBUNG NUMERIC
@ATTRIBUTE jml_KT_PENEGASAN NUMERIC
@ATTRIBUTE jml_KT_PASIF NUMERIC
@ATTRIBUTE jml_KT_NEGATIF NUMERIC
@ATTRIBUTE Class {Incomplete,Complete}

@DATA
% ---from Corpus_Incomplete
2, 0, 0, 0, 18, 1, 2, 1, 0, 0, Incomplete
3, 0, 0, 0, 29, 3, 2, 1, 0, 0, Incomplete
3, 0, 0, 1, 22, 5, 1, 0, 1, 0, Incomplete
...
...

%-----from corpus Complete
0, 0, 0, 0, 7, 1, 1, 0, 0, 0, Complete
0, 0, 0, 0, 10, 1, 0, 1, 0, 0, Complete
1, 0, 0, 0, 10, 1, 1, 1, 0, 0, Complete
...
...
```

Gambar 4.2 Data format ARFF Hasil Ekstraksi Fitur

4.2.3.3. Hasil Pembangkitan Data Buatan dengan Algoritma SMOTE

Tahap pembangkitan data buatan dengan Algoritma SMOTE berfungsi untuk memperbanyak data dari kelas yang minoritas agar sebanding dengan kelas mayoritas. Data yang terbentuk dari kesepakatan para ahli 427 terdiri dari 141 corpus berlabel tidak lengkap dan 286 corpus berlabel lengkap, Setelah dilakukan proses pembangkitan data dengan SMOTE maka dihasilkan data sejumlah 568 terdiri dari 282 corpus berlabel tidak lengkap dan 286 corpus berlabel lengkap. Terdapat penambahan sebanyak 141 corpus berlabel tidak lengkap. Pembangkitan data dengan metode SMOTE menghasilkan data berupa bilangan decimal seperti yang ditunjukkan pada gambar 4.3. Oleh karena itu, perlu dirubah menjadi bentuk bilangan bulat seperti yang ditunjukkan pada gambar 4.4 merupakan hasil perubahannya.

```
. . .  
2,0,0,0,18.71074,1,1.840422,0.445928,0,0,Incomplete  
1.989316,0,0,0,10,2,0,0.782398,0,0,Incomplete  
2,0,0,0,10,1.586248,0.037785,1,0,0,Incomplete  
1,0,0,0.198843,14,2.481213,0,0,0,0,Incomplete  
0,0,0,0,15.836896,2,0.99124,0,0,0,Incomplete  
. . .
```

Gambar 4.3 Data Buatan Hasil Pembangkitan Metode SMOTE

```
. . .  
2,0,0,0,19,1,2,1,0,0,Incomplete  
2,0,0,0,10,2,0,1,0,0,Incomplete  
2,0,0,0,10,2,1,1,0,0,Incomplete  
1,0,0,1,14,3,0,0,0,0,Incomplete  
0,0,0,0,16,2,1,0,0,0,Incomplete  
. . .
```

Gambar 4.4 Data Buatan Hasil Pembangkitan Metode SMOTE

4.2.3.4. Perankingan Fitur Berdasarkan *Information Gain*

Tabel 4.2 berikut ini adalah daftar semua fitur yang dipergunakan untuk mendeteksi ketidaklengkapan serta menunjukkan hasil perankingan

dengan metode Information Gain. Dari tabel 4.2 dapat diketahui bahwa semakin tinggi peringkatnya maka fitur tersebut semakin berpengaruh terhadap hasil klasifikasi. Kolom sebelah kiri dari tabel 4.2 merupakan perankingan fitur dengan menggunakan data sebelum dilakukan metode SMOTE, dan sebelah kanan merupakan perankingan dengan menggunakan data hasil SMOTE.

Tabel 4.2 Tabel Perankingan Fitur

NO	Tanpa SMOTE		Dengan SMOTE	
	Fitur	Nilai	Fitur	Nilai
1	Kata sifat ketidaklengkapan (bad_jj)	0.09133	Kata sifat ketidaklengkapan (bad_jj)	0.14581
2	Kata keterangan ketidaklengkapan (bad_rb)	0.03432	Kata keterangan ketidaklengkapan (bad_rb)	0.05451
3	Jumlah kata penegasan	0.01938	Jumlah kata penegasan	0.04555
4	Jumlah kata penghubung	0.00170	Modal ketidaklengkapan (bad_md)	0.00778
5	Kata ganti ketidaklengkapan (bad_prp)	0.00126	Kata ganti ketidaklengkapan (bad_prp)	0.00386
6	Jumlah kata negatif	0.00103	Jumlah kata kerja pasif	0.00187
7	Modal ketidaklengkapan (bad_md)	0.00065	Jumlah kata negatif	0.00047
8	Jumlah kata kerja pasif	0.00025	Jumlah kata kerja	0.00043
9	Jumlah kata kerja	0.000002	Jumlah kata penghubung	0.0003
10	Jumlah kata	0	Jumlah kata	0

4.2.3.5. Pendeteksian Ketidاكلengkapan dengan Semua Fitur

Pengujian pendeteksian ketidaklengkapan dengan menggunakan metode *crossvalidation* sebanyak 10 fold. Penggunaan metode *crossvalidation* 10 fold berarti data akan dipilah sebanyak 10 bagian, 9 bagian akan digunakan sebagai latih dan 1 bagian sebagai uji, begitu seterusnya sampai setiap bagian pernah sebagai uji.

Skenario pengujian pertama yaitu menggunakan semua fitur yang ada pada tabel 4.2 untuk diklasifikasi dengan kombinasi metode Adaboost dan C4.5. Data yang digunakan adalah data yang tidak seimbang jumlah kelasnya (sebelum proses SMOTE) dan data hasil pembangkitan (proses SMOTE). Pada

tahap ini kedua data akan diujikan dengan 10 iterasi, 20 iterasi, 40 iterasi, 70 iterasi, 100 iterasi, dan 130 iterasi. Pengujian pertama ini bertujuan mengetahui jumlah iterasi yang tepat. Tabel 4.3 merupakan hasil pengujian beberapa iterasi dengan kombinasi metode adaboost dan C4.5.

Tabel 4.3 Tabel Hasil Klasifikasi dengan Kombinasi Metode Adaboost dan C4.5

Jml Iterasi	Data Asli		Data sesudah SMOTE	
	Confusion Matric	Gwet AC1 Static	Confusion Matric	Gwet AC1 Static
10	a b <-- classified as 65 76 a = Incomplete 64 222 b = Complete	0.42	a b <-- classified as 216 66 a = Incomplete 76 210 b = Complete	0.50
20	a b <-- classified as 64 77 a = Incomplete 65 221 b = Complete	0.41	a b <-- classified as 218 64 a = Incomplete 76 210 b = Complete	0.51
40	a b <-- classified as 65 76 a = Incomplete 64 222 b = Complete	0.42	a b <-- classified as 218 64 a = Incomplete 74 212 b = Complete	0.51
70	a b <-- classified as 65 76 a = Incomplete 64 222 b = Complete	0.42	a b <-- classified as 218 64 a = Incomplete 74 212 b = Complete	0.51
100	a b <-- classified as 65 76 a = Incomplete 64 222 b = Complete	0.42	a b <-- classified as 218 64 a = Incomplete 74 212 b = Complete	0.51
130	a b <-- classified as 65 76 a = Incomplete 64 222 b = Complete	0.42	a b <-- classified as 218 64 a = Incomplete 74 212 b = Complete	0.51

Dari tabel 4.3 dapat diketahui bahwa hasil stabil dari kedua data tersebut dimulai pada iterasi ke-40 yang ditunjukkan indeks kesepakatan 0.41(data asli) dan 0.51(data setelah SMOTE). Jumlah iterasi juga menunjukkan jumlah C4.5 yang digunakan oleh adaboost. Iterasi diperlukan untuk menemukan bobot dan banyaknya C4.5 yang digunakan. Berdasarkan tabel 4.3 menunjukkan indeks kesepakatan untuk data setelah SMOTE lebih tinggi berada pada tingkat *moderate*/sedang (0.51) dan data asli juga pada tingkat *moderate* (0.41).

4.2.3.6. Pendeteksian Ketidaklengkapan dengan Beberapa Fitur Tertentu

Proses ini digunakan untuk mengetahui fitur apasaja yang berpengaruh pada klasifikasi. Dengan menerapkan hasil perankingan fitur sesuai tabel 4.2 pada klasifikasi dengan kombinasi metode adaboost dan C4.5 menggunakan iterasi sebanyak 40. Penerapan fitur sesuai dengan hasil metode *information gain* (ditunjukkan pada tabel 4.2), menggunakan 9 fitur teratas, 8 fitur teratas, 6 fitur teratas, 5 fitur teratas, dan 4 fitur teratas. Tabel 4.4 merupakan hasil klasifikasi dengan sejumlah fitur teratas, yaitu dari 9 fitur teratas sampai 4 fitur teratas.

Tabel 4.4 Tabel Hasil Klasifikasi dengan Sejumlah Fitur Tertentu

Jml Fitur	Data Asli		Data sesudah SMOTE	
	Confusion Matrix	Gwet AC1 Static	Confusion Matrix	Gwet AC1 Static
9	a b <-- classified as 63 78 a = Incomplete 63 223 b = Complete	0.42	a b <-- classified as 212 70 a = Incomplete 79 207 b = Complete	0.48
8	a b <-- classified as 60 81 a = Incomplete 51 235 b = Complete	0.47	a b <-- classified as 213 69 a = Incomplete 74 212 b = Complete	0.50
7	a b <-- classified as 51 90 a = Incomplete 41 245 b = Complete	0.49	a b <-- classified as 209 73 a = Incomplete 78 208 b = Complete	0.47
6	a b <-- classified as 55 86 a = Incomplete 38 248 b = Complete	0.52	a b <-- classified as 211 71 a = Incomplete 89 197 b = Complete	0.44
5	a b <-- classified as 55 86 a = Incomplete 44 242 b = Complete	0.49	a b <-- classified as 206 76 a = Incomplete 90 196 b = Complete	0.42
4	a b <-- classified as 50 91 a = Incomplete 36 250 b = Complete	0.51	a b <-- classified as 232 50 a = Incomplete 112 174 b = Complete	0.44

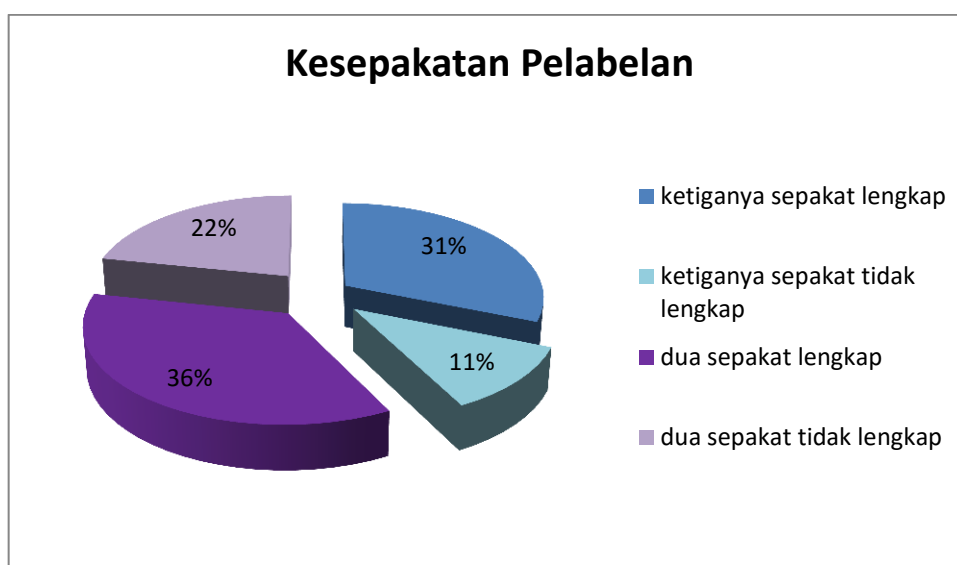
Dari hasil tabel 4.4 dapat diketahui bahwa 6 fitur teratas dari data tanpa hasil SMOTE sangat berpengaruh terhadap hasil klasifikasi, sedangkan untuk data dengan SMOTE 8 fitur teratasnya yang sangat berpengaruh. Terbukti dengan hasil akurasi yang terus naik sampai pada fitur 6 teratas untuk

data tanpa SMOTE dan pada fitur 8 teratas untuk data dengan SMOTE. Tabel 4.4 juga menunjukkan indeks kesepakatan yang tertinggi dari masing-masing adalah 0.52 pada data sebelum SMOTE dan 0.5 pada data setelah SMOTE.

4.3. Analisa Hasil

4.3.1. Analisa Kesepakatan Antar Ahli dan Ahli dengan Kerangka Kerja

Setiap data yang dilabeli oleh tiga orang ahli, suara terbanyak akan menentukan data tersebut lengkap atau tidak lengkap. Hasil pelabelan dari ketiga ahli pada masing-masing data ditunjukkan pada gambar 4.5. Pada gambar 4.5 menunjukkan bahwa dari total 427 data ahli lebih banyak sepakat berlabel lengkap dengan perolehan 31% ketiganya berpendapat sama dan 36% dua ahli yang berpendapat sama, sedangkan berlabel tidak lengkap hanya 11% ketiga ahli berpendapat sama dan 22% dua ahli berpendapat sama.



Gambar 4.5 Diagram Distribusi Kesepakatan Ahli

Tabal 4.5 merupakan tabel perbandingan kesepakatan diantara ahli. Dengan membandingkan hasil labeling diantara ahli maka dapat diketahui bahwa kesepakatan tertinggi antara ahli 2 dan ahli 3 sedangkan kesepatan terendah antara ahli 1 dengan ahli 2.

Tabel 4.5 Tabel Perbandingan Kesepakatan Antar Ahli

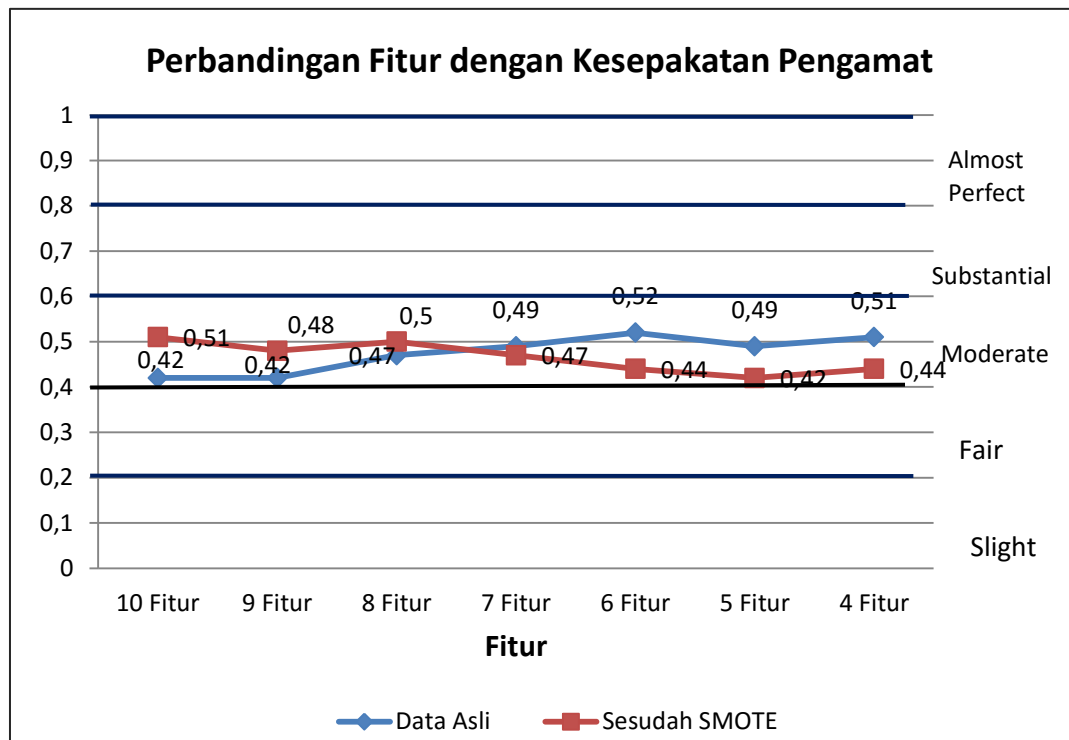
	Kerangka kerja	Ahli 1	Ahli 2	Ahli 3
Kerangka kerja	*	0.54	0.65	0.71
Ahli 1	0.54	*	0.250850603	0.313976
Ahli 2	0.65	0.250850603	*	0.398606
Ahli 3	0.71	0.313976	0.398606	*

Dari table 4.5 dapat dilihat bahwa indeks kesepakatan antara ahli dengan kerangka kerja lebih bagus dari antar ahli sendiri dengan indeks kesepakatan terendah oleh ahli 1, sedangkan dua ahli lain memiliki hasil *substantial*. Tabel 4.5 juga menunjukkan ketiga ahli ketiga dibandingkan dengan menggunakan 10 fitur rata-rata berada pada tingkat *moderate* untuk ahli ke 1, sedangkan ahli ke 2 dan ahli ke 3 nilainya mencapai *substansial*.

Dari tabel 4.5 dapat diketahui bahwa kerangka kerja memberikan nilai yang lebih baik dari pelabelan manual. Terbukti pelabelan manual (ahli) nilai kesepakatan rata-rata pada tingkat *fair* sedangkan dengan kerangka kerja nilai kesepakatan rata-rata pada tingkat *moderate* sampai *substantial*.

4.3.2. Analisa Penggunaan Fitur Terhadap Klasifikasi Ketidaklengkapan Kebutuhan

Penggunaan metode *information gain* dalam melakukan perankingan fitur terbukti dapat meningkatkan hasil klasifikasi. Hasil uji coba yang tunjukkan pada subbab 4.2.3.5 (tabel 4.3) dan subbab 4.2.3.6 (tabel 4.4) menunjukkan hasil kappa dengan Gwet AC-1 yang berbeda. Hasil tabel 4.4 menunjukkan indeks kesepakatan kedua data pada tingkat *moderate*.



Gambar 4.6 Grafik Hasil Evaluasi Fitur

Grafik pada gambar 4.7 diperoleh dengan menerapkan iterasi sebanyak 40 (hasil percobaan subbab 4.2.3.6) terhadap data uji pada beberapa fitur tertentu. Pada gambar 4.7 menunjukkan penggunaan data setelah SMOTE awalnya lebih tinggi bila dibandingkan dengan menggunakan data tanpa proses SMOTE, akan tetapi setelah penggunaan 7 fitur teratas hasilnya semakin menurun. Sedangkan pada data asli terus meningkat sampai pada 6 fitur teratas, selanjutnya pada penggunaan 5 fitur teratas mengalami penurunan.

Maka penggunaan jumlah fitur yang optimal untuk data asli yaitu pada 6 fitur teratas antara lain: kata sifat ketidaklengkapan (BAD_JJ), kata keterangan ketidaklengkapan (BAD_RB), jumlah kata penegasan, jumlah kata penghubung, kata ganti ketidaklengkapan (BAD_PRP), dan jumlah kata negative. Sedangkan untuk data dengan proses SMOTE fitur optimal dengan 8 fitur teratas antara lain: kata sifat ketidaklengkapan (BAD_JJ),), kata keterangan ketidaklengkapan (BAD_RB), *modal* ketidaklengkapan

(BAD_MD), kata ganti ketidaklengkapan (BAD_PRP), jumlah kata pasif, jumlah kata negatif, jumlah kata penegasan, dan jumlah kata kerja.

4.3.3. Analisa Salah Prediksi oleh Kerangka Kerja

Untuk menganalisa data-data yang salah prediksi oleh kerangka kerja. Maka dilakukan uji coba dengan membagi data menjadi data uji dan data latih. Terdapat dua data latih dan satu data uji, data latih pertama berasal dari data asli berisi 120 data berlabel tidak lengkap dan 120 data berlabel lengkap. Data latih kedua merupakan kombinasi dari 60 data berlabel tidak lengkap data asli dan 60 data berlabel tidak lengkap hasil SMOTE, serta 120 berlabel lengkap. Untuk data uji total sebanyak 328 terdiri dari 162 data berlabel tidak lengkap kombinasi data asli dan buatan serta 166 data berlabel lengkap.

Kedua data latih dilakukan proses klasifikasi sehingga menghasilkan dua model klasifikasi. Selanjutnya kedua model klasifikasi tersebut diujicobakan pada data uji yang sama.

Tabel 4.6 merupakan tabel perbandingan dari hasil *training* dan *testing* untuk menghasilkan analisa berikut. Dari tabel 4.6 juga diketahui bahwa dengan menggunakan SMOTE maka memperoleh hasil yang lebih baik.

Tabel 4.6 Tabel Perbandingan Hasil

Data Asli		Data Buatan (Dengan SMOTE)	
Akurasi	Gwet AC1	Akurasi	Gwet AC1
71.95%	0.453	76.22%	0.525

Detail dari kesalahan prediksi oleh dua model kerangka kerja ditunjukkan pada Tabel 4.7, Tabel 4.8 dan Tabel 4.9. Tabel 4.7, Tabel 4.8 dan Tabel 4.9 merupakan beberapa pernyataan kebutuhan yang berbeda kesepakatan antara ahli dengan kerangka kerja. Dengan menerapkan rule model yang telah terbentuk, selanjutnya diujikan pada data uji. Tabel 4.6 merupakan kelompok data yang diprediksi oleh ahli sebagai tidak lengkap, tabel 4.7 merupakan kelompok data yang diprediksi oleh ahli sebagai lengkap dan tabel 4.8 merupakan kelompok data buatan dengan label tidak lengkap.

Tabel 4.7 Tabel Pernyataan Kebutuhan yang Beda Kesepakatan Antara Pelabelan Ahli dengan Kerangka Kerja (Menurut Ahli *Incomplete*)

Kebutuhan Ke-	Detail	Ahli	Dengan Kerangka Kerja	
			Tanpa SMOTE	Dengan SMOTE
121	An efficient mechanism to backup and recover the data this may include a distributed backup or centralized backup.	<i>Incomplete</i>	<i>Complete</i>	<i>Complete</i>
	[an_DT, efficient_JJ, mechanism_NN, to_TO, backup_VB, and_CC, recover_VB, the_DT, data_NNS, this_DT, may_MD, include_VB, a_DT, distributed_VBN, backup_NN, or_CC, centralized_JJ, backup_NN]			
127	This report lists the status of all expense reports over a specific period of time	<i>Incomplete</i>	<i>Incomplete</i>	<i>Complete</i>
	[this_DT, report_NN, lists_VBZ, the_DT, status_NN, of_IN, all_DT, expense_NN, reports_NNS, over_IN, a_DT, specific_JJ, period_NN, of_IN, time_NN]			
130	PID loop maintains thresholds so that spec is met	<i>Incomplete</i>	<i>Complete</i>	<i>Complete</i>
	[pid_NNP, loop_NN, maintains_VBZ, thresholds_NNS, so_IN, that_IN, spec_NN, is_VBZ, met_VBN]			
132	It is expected that the system has an administrator interface where users can be created updated with appropriate authorities	<i>Incomplete</i>	<i>Incomplete</i>	<i>Complete</i>
	[it_PRP, is_VBZ, expected_VBN, that_IN, the_DT, system_NN, has_VBZ, an_DT, administrator_NN, interface_NN, where_WRB, users_NNS, can_MD, be_VB, created_VBN, updated_VBN, with_IN, appropriate_JJ, authorities_NNS]			
136	They call the help desk to actualize the changes	<i>Incomplete</i>	<i>Complete</i>	<i>Complete</i>
	[they_PRP, call_VBP, the_DT, help_NN, desk_NN, to_TO, actualize_VB, the_DT, changes_NNS]			
137	The system will work in a proper speed so the required time for study will be decreased	<i>Incomplete</i>	<i>Complete</i>	<i>Incomplete</i>
	[the_DT, system_NN, will_MD, work_VB, in_IN, a_DT, proper_JJ, speed_NN, so_IN, the_DT, required_VBN, time_NN, for_IN, study_NN, will_MD, be_VB, decreased_VBN]			
140	The coloring and contrast adjustment helps them concentrate on a specific group of cells	<i>Incomplete</i>	<i>Complete</i>	<i>Complete</i>
	[the_DT, coloring_NN, and_CC, contrast_NN, adjustment_NN, helps_VBZ, them_PRP,			

	concentrate_VB, on_IN, a_DT, specific_JJ, group_NN, of_IN, cells_NNS]			
--	---	--	--	--

Tabel 4.8 Tabel Pernyataan Kebutuhan yang Beda Kesepakatan Antara Pelabelan Ahli dengan Kerangka Kerja (Menurut Ahli *Complete*)

Kebutuhan an Ke-	Detail	Ahli	Dengan Kerangka Kerja	
			Tanpa SMOTE	Dengan SMOTE
263	All members who log on after the message is posted can view this message and reply to it if they want [all_DT, members_NNS, who_WP, log_VBP, on_IN, after_IN, the_DT, message_NN, is_VBZ, posted_VBN, can_MD, view_VB, this_DT, message_NN, and_CC, reply_NN, to_TO, it_PRP, if_IN, they_PRP, want_VBP]	<i>Complete</i>	<i>Complete</i>	<i>Incomplete</i>
269	They can either chat with individual members of the group or they can chat in a Conference with more than one member [they_PRP, can_MD, either_RB, chat_VB, with_IN, individual_JJ, members_NNS, of_IN, the_DT, group_NN, or_CC, they_PRP, can_MD, chat_VB, in_IN, a_DT, conference_NN, with_IN, more_JJR, than_IN, one_CD, member_NN]	<i>Complete</i>	<i>Complete</i>	<i>Incomplete</i>
270	Only the owner of the group can add members to the group [only_RB, the_DT, owner_NN, of_IN, the_DT, group_NN, can_MD, add_VB, members_NNS, to_TO, the_DT, group_NN]	<i>Complete</i>	<i>Incomplete</i>	<i>Complete</i>
271	The user goes to www MyFamilyReunion com to create a new group family [the_DT, user_NN, goes_VBZ, to_TO, to_TO, create_VB, a_DT, new_JJ, group_NN, family_NN]	<i>Complete</i>	<i>Incomplete</i>	<i>Complete</i>
273	To post a message, the member clicks on the post message link [to_TO, post_VB, a_DT, message_NN, the_DT, member_NN, clicks_VBZ, on_IN, the_DT, post_NN, message_NN, link_NN]	<i>Complete</i>	<i>Incomplete</i>	<i>Incomplete</i>
274	The users can also reply to a posted message [the_DT, users_NNS, can_MD, also_RB, reply_VB, to_TO, a_DT, posted_VBN, message_NN]	<i>Complete</i>	<i>Incomplete</i>	<i>Incomplete</i>
279	An administrator adds a new therapist	<i>Complete</i>	<i>Incomplete</i>	<i>Incomplete</i>

	[an_DT, administrator_NN, adds_VBZ, a_DT, new_JJ, therapist_NN]			
280	An administrator adds a user to the system who is a parent	<i>Complete</i>	<i>Incomplete</i>	<i>Incomplete</i>
	[an_DT, administrator_NN, adds_VBZ, a_DT, user_NN, to_TO, the_DT, system_NN, who_WP, is_VBZ, a_DT, parent_NN]			

Tabel 4.9 Tabel Data Buatan yang Salah Prediksi

No	Kt sifat	Kt keterangan	Kt ganti	Modal	Jml kata	Jml_kt_kerja	Jml_kt_penghubung	Jml_kt_penegasan	Jml kt pasif	Jml_kt_negatif	Kelas	Kerang ka Kerja	
												Tanpa SMOTE	Dengan SMOTE
489	1	0	1	0	20	2	0	0	0	0	Incomplete	Incomplete	Complete
491	1	0	0	0	10	2	0	0	0	0	Incomplete	Incomplete	Complete
496	2	0	0	0	12	1	1	0	1	0	Incomplete	Incomplete	Complete
500	1	0	0	0	19	2	1	0	0	0	Incomplete	Complete	Complete
503	1	0	0	0	14	1	2	0	0	0	Incomplete	Incomplete	Complete
515	1	2	2	1	41	4	2	1	1	0	Incomplete	Complete	Complete
519	1	0	0	0	14	2	0	0	0	0	Incomplete	Incomplete	Complete
522	1	0	0	1	14	2	0	0	0	0	Incomplete	Complete	Incomplete
531	1	0	1	1	14	2	0	0	0	0	Incomplete	Complete	Complete
535	0	0	0	0	7	1	1	0	0	0	Incomplete	Complete	Complete
541	2	1	0	0	35	5	2	0	1	2	Incomplete	Complete	Incomplete
549	1	0	0	0	15	2	0	0	0	0	Incomplete	Complete	Complete

Dari Tabel 4.7, Tabel 4.8 dan Tabel 4.9 maka dianalisa dengan mencocokkan rule model klasifikasi yang dihasilkan pada saat latih, maka diambil beberapa rule yang mempengaruhi salah prediksi. Berdasarkan rule tersebut maka dapat diketahui bahwa fitur yang paling dominan adalah kata sifat ketidaklengkapan, kata keterangan ketidaklengkapan, kata ganti ketidaklengkapan, jml_kata_penghubung, dan jml_kt_kerja_pasif untuk kerangka kerja tanpa SMOTE. Sedangkan untuk kerangka kerja dengan SMOTE fitur yang dominan antara lain kata sifat ketidaklengkapan, kata keterangan ketidaklengkapan, kata ganti ketidaklengkapan, jml_kt_penegasan, jml_kt_pasif, dan modal ketidaklengkapan. Fitur-fitur tersebut sesuai dengan hasil perankingan, dimana fitur optimal dari kerangka kerja tanpa SMOTE merupakan fitur yang masuk dalam 6 fitur teratas, kecuali jml_kt_kerja pasif. Fitur optimal kerangka kerja dengan SMOTE juga termasuk 8 fitur teratas yang memberikan nilai bagus pada percobaan di Gambar 4.7.

[Halaman ini sengaja dikosongkan]

BAB 5

PENUTUP

Pada bab ini dijelaskan mengenai kesimpulan akhir yang didapat setelah melakukan serangkaian uji coba pada bab sebelumnya.

5.1. Kesimpulan

Kesimpulan yang dapat diambil dalam penelitian ini antara lain adalah sebagai berikut.

- a. Penggunaan metode *information gain* terbukti efektif untuk perankingan data. Terbukti 6 fitur teratasnya untuk kerangka kerja tanpa SMOTE mampu meningkatkan hasil klasifikasi sebesar 0,52 (*moderate*) dan 8 fitur teratas untuk kerangka kerja dengan SMOTE sebesar 0,5 (*moderate*).
- b. 6 Fitur teratas untuk kerangka kerja tanpa SMOTE antara lain BAD_JJ, BAD_RB, jumlah kata penegasan, jumlah kata penghubung, BAD_PRP, dan jumlah kata negative. Sedangkan 8 fitur teratas untuk kerangka kerja dengan SMOTE antara lain: BAD_JJ, BAD_RB, jumlah kata penegasan, bad_MD, bad_PRP, jumlah kata negative, dan jumlah kata pasif.
- c. Penggunaan kombinasi algoritma adaboost dan C4.5 mampu mendeteksi ketidaklengkapan pada rata-rata tingkat *moderate*.
- d. Indeks kesepakatan antara ahli dengan kerangka kerja rata-rata berada pada tingkat *moderate*, ini lebih tinggi bila dibandingkan dengan indeks kesepakatan antar ahli sendiri yang hanya rata-rata pada tingkat *fair*.

5.2. Saran

Saran yang dapat diberikan untuk pengembangan lebih lanjut penelitian ini adalah sebagai berikut.

- Penelitian menggunakan dokumen SKPL berbahasa Inggris untuk proses ujicoba. Untuk penelitian selanjutnya disarankan untuk menggunakan bahasa lain seperti Bahasa Indonesia.
- Pembentukan *corpus* pada penelitian ini masih sepenuhnya manual. Untuk penelitian selanjutnya disarankan untuk menggunakan menambahkan

proses otomatis atau semi otomatis untuk pembentukan *corpus* sehingga mampu mengurangi subjektivitas dari para ahli.

- Memperbanyak data khususnya untuk jumlah pernyataan kebutuhan pada corpus tidak lengkap.
- Memperbaiki proses penentuan fitur yang dilibatkan dalam pendeteksian ketidaklengkapan sehingga dapat mengurangi jumlah ketidak sepakatan antara kerangka kerja dan ahli.
- Menambahkan semantik dalam pendeteksian ketidaklengkapan pada dokumen SKPL.

DAFTAR PUSTAKA

- Barro, R.A., Sulvianti, I.D. & Afendi, F.M., 2013. PENERAPAN SYNTHETIC MINORITY OVERSAMPLING TECHNIQUE. , 1(1).
- Carlson, N. & Laplante, P., 2014. The NASA automated requirements measurement tool: a reconstruction. *Innovations Syst Softw Eng*, 10, pp.77–91.
- Chawla, N. V et al., 2002. SMOTE: Synthetic Minority Over-sampling Technique. , 16, pp.321–357.
- Fabbrini, F. et al., 2000. Quality evaluation of software requirement specifications. *Proceedings of the Software and Internet Quality Week 2000 Conference*, pp.1–18. Available at: <http://fmt.isti.cnr.it/WEBPAPER/paper8A2.pdf>.
- Firesmith, D., 2005. Are Your Requirements Complete ? , 4(1), pp.27–43.
- Genova, G. et al., 2013. A framework to measure and improve the quality of textual requirements. *Requirements Engineering*, 18(1), pp.25–41.
- Gralha, C., Araújo, J. & Goulão, M., 2015. Metrics for measuring complexity and completeness for social goal models. *Information Systems*, 53, pp.346–362. Available at: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84933674144&partnerID=tZOtx3y1>.
- Gwet, K., 2002. Kappa Statistic is not Satisfactory for Assessing the Extent of Agreement Between Raters. , (1), pp.1–5.
- Hotho, A., Nürnberger, A. & Paaß, G., 2005. A Brief Survey of Text Mining. *LDV Forum - GLDV Journal for Computational Linguistics and Language Technology*, 20, pp.19–62. Available at: <http://www.kde.cs.uni-kassel.de/hotho/pub/2005/hotho05TextMining.pdf>.
- Hussain, I., Ormandjieva, O. & Kosseim, L., 2007. Automatic quality assessment of SRS text by means of a decision-tree-based text classifier. In *Proceedings - International Conference on Quality Software*. pp. 209–218.
- INCOSE, R.W.G., 2015. *Guide for Writing Requirements*, Available at: <http://www.incose.org/ProductsPublications/techpublications/GuideRequirements>.
- ISO/IEC/ IEEE 29148, 2011. INTERNATIONAL STANDARD ISO / IEC / IEEE Systems and software engineering — engineering. *Iso/Iec/Ieee 29148:2011(E)*, 2011, pp.1–94.
- Kiyavitskaya, N. et al., 2008. Requirements for tools for ambiguity identification and measurement in natural language requirements specifications. *Requirements Engineering*, 13(3), pp.207–239.
- Landis, J.R. & Koch, G.G., 2012. The Measurement of Observer Agreement for Categorical Data Data for Categorical of Observer Agreement The Measurement. , 33(1), pp.159–174.
- Lee, C. & Lee, G.G., 2006. Information gain and divergence-based feature selection for machine learning-based text categorization. *Information Processing and Management*, 42(1 SPEC. ISS), pp.155–165.

- Parra, E. et al., 2015. A methodology for the classification of quality of requirements using machine learning techniques. *Information and Software Technology*, 67, pp.180–195. Available at: <http://dx.doi.org/10.1016/j.infsof.2015.07.006>.
- Popescu, D. et al., 2008. Reducing ambiguities in requirements specifications via automatically created object-oriented models. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 5320 LNCS, pp.103–124.
- Pramanik, S. et al., 2010. A Comparative Study of Bagging , Boosting and C4 . 5 : The Recent Improvements in Decision Tree Learning Algorithm. *Asian Journal of Information Technology*, (June 2010), pp.300–306.
- Pressman, R., 2012. *Rekayasa Perangkat Lunak* 7th ed., Yogyakarta: Andi.
- Quinlan, J.R., 1992. *C4.5: Programs for Machine Learning*,
- Rokach, L., 2005. Ensemble Methods in Supervised Learning. In *Data Mining and Knowledge Discovery Handbook*. pp. 959–975. Available at: http://link.springer.com/10.1007/0-387-25465-X_2.
- Salzberg, S.L., 1994. Book Review: C4. 5: programs for machine learning. *Machine Learning*, 240, p.302.
- Sardinha, A. et al., 2013. EA-Analyzer: Automating conflict detection in a large set of textual aspect-oriented requirements. *Automated Software Engineering*, 20(1), pp.111–135.
- Siahaan, D., 2012. *Analisa Kebutuhan Dalam Rekayasa Perangkat Lunak*, Yogyakarta: Andi.
- de Sousa, T.C. et al., 2010. Automatic analysis of requirements consistency with the B method. *ACM SIGSOFT Software Engineering Notes*, 35(2), p.1.
- Vie, D.L.J., 2010. *writing-software-requirements-specifications*, Available at: <http://www.letu.edu/people/jaytevis/Software-Engineering/Examples/writing-software-requirements-specifications.doc>.

LAMPIRAN

Tabel Daftar Pernyataan Kebutuhan yang Telah Dianotasi

KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM001001	The framework must be scalable for several hundred case workers and tens of thousands of cases and clients.	incomplete	incomplete	incomplete
ACM001002	Response times for case workers must be less than two seconds on saving and retrieving case information, and sub-second on moving between views of different data for the current case.	incomplete	incomplete	complete
ACM001003	Line Management requires daily and weekly reporting.	complete	complete	incomplete
ACM001004	Security is needed to control which users can do certain actions, such as assigning cases, approving a request, or editing particular fields.	incomplete	incomplete	complete
ACM001005	The system must support multi-lingual deployment using multi-byte character encoding.	incomplete	complete	incomplete
ACM001006	The system must include an audit trail for each case.	complete	complete	incomplete
ACM001007	The system must include archiving of old or inactive data.	complete	incomplete	incomplete
ACM001008	The user should be presented with a list of matching entries, with a variety of specific details to help choose the desired case.	incomplete	incomplete	incomplete
ACM001009	When entering a name the system must search for all matches and allow the case worker to choose one from those found, or create a new entry.	complete	complete	complete
ACM001010	Note that the tombstone data will vary for different applications developed using the framework.	incomplete	incomplete	incomplete
ACM001011	Report an error if mandatory information is missing.	complete	complete	incomplete
ACM001012	Edit the status of a case, automatically creating any events, or bring-forwards caused by the status change.	incomplete	complete	incomplete
ACM001013	The system should be able to automatically create or edit a BF based on an event or a status change.	incomplete	complete	complete
ACM001014	Depending on the result, the case may remain with the worker, or be forwarded along various paths according to the configuration for this application of the framework.	incomplete	incomplete	incomplete
ACM001015	The system should be configurable to support multiple approval/rejection points for the lifecycle of a case.	complete	complete	incomplete

KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM001016	A case may be rejected at one point and this could trigger a variant of a request, or a case lifecycle might consist of a single approval/rejection.	incomplete	incomplete	incomplete
ACM001017	Search and retrieve the event entries for the currently selected case.	complete	complete	complete
ACM001018	Search the bring-forward entries using any combination of the following criteria: case, case worker, status, or date range.	complete	complete	complete
ACM001019	A manager can do any of the use-cases for Case Workers.	incomplete	complete	incomplete
ACM001020	Assign a case to a specific case worker or to a pool of case workers of the same subtype.	incomplete	incomplete	incomplete
ACM001021	View the workload of case workers reporting to them.	incomplete	incomplete	incomplete
ACM001022	The system should be configurable as to whether history is maintained or not for client/tombstone data management.	complete	complete	complete
ACM001023	The framework should not preclude changes by the company as to the lifecycle of a case.	complete	complete	complete
ACM001024	Similar to the above, the framework should not preclude changes to the workflow.	complete	incomplete	incomplete
ACM002001	Adding new cells and new cell types should be relatively easy	incomplete	incomplete	incomplete
ACM002002	The operator must be able to plan to change every parameter for every cell.	complete	complete	incomplete
ACM002003	It must be possible to upgrade a cell to a cell of another type and revert back, defaults should be assigned to new parameters.	incomplete	incomplete	incomplete
ACM002004	All parameters with the same name should be converted to the new type without loss of information.	complete	incomplete	complete
ACM002005	It can of course be assumed that the program that does the conversion knows the new and old cell types.	incomplete	incomplete	incomplete
ACM002006	It must be possible to develop "small" software modules that can work upon the information without getting intricately tied to all the possible cell types.	incomplete	incomplete	incomplete
ACM002007	When a change in the network fails, the plan should be rolled back to bring the network back to its former state	complete	complete	complete
ACM002008	The program should have no restrictions on the number of cell types, or number of cells.	complete	complete	complete
ACM002009	It should be possible to minimize the number of commands sent to the cells.	incomplete	incomplete	incomplete
ACM002010	The reason is that certain cells have a very slow connection (seconds per command).	incomplete	incomplete	incomplete

KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM002011	The original cell was a Sienokola and used an antenna that created a cell where the site was in the middle.	incomplete	incomplete	complete
ACM002012	A new supplier has entered the market that can supply base stations for a significantly lower price than the ones currently used.	incomplete	incomplete	incomplete
ACM002013	The network operator decides to buy 10 base stations and installs them on new sites.	complete	incomplete	incomplete
ACM003001	Allow customers to view products from their home	complete	complete	complete
ACM003002	Allow customers to make purchases and receive products from their home.	complete	complete	incomplete
ACM003003	Customers are able to make payments from credit card.	complete	complete	complete
ACM003004	Product descriptions must be easily maintained and updated by store employees	incomplete	incomplete	incomplete
ACM003005	Fast distribution of product descriptions so that customers have accurate portrayal of what Doe's Electronics has to offer	incomplete	incomplete	incomplete
ACM003006	Inexpensive distribution of product descriptions	incomplete	incomplete	incomplete
ACM003007	Fast customer ordering system that is as simple as possible	incomplete	incomplete	incomplete
ACM003008	The customer can login to the e-Commerce shopping system by enter his user name and password.	complete	complete	complete
ACM003009	When the customer finds the products he wants, he adds them to the shopping carts.	complete	complete	incomplete
ACM003010	The customer can request to view the contents of the shopping cart.	complete	complete	complete
ACM003011	The customer can request to update their customer info.	complete	complete	complete
ACM003012	If the customer is a new user, he can request to register with the system.	complete	complete	complete
ACM003013	When the customer requests to checkout and he does not have credit card information stored at this point (system can not find his payment information), the system will prompt credit card information page.	complete	complete	complete
ACM003014	When the customer checks out, the credit verification company validates the customer's credit card when given the customers name, credit card number, and expiration date, and then returns the validation result to checkout department (sales person).	complete	complete	complete
ACM003015	The sales clerk requests to update products information.	complete	complete	complete
ACM003016	The sales clerk or shipping clerk requests to update inventory.	complete	complete	complete
ACM003017	After getting the order request, the sales clerk ships the order products to the customer within three to five business days.	complete	complete	complete

ACM003018	When the customer finishes shopping, he requests to checkout.	complete	complete	complete
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM003019	The customer requests to view the products in a product category.	complete	complete	complete
ACM003020	The customer enters product search parameters and requests a product search.	complete	complete	incomplete
ACM004001	All event objects must contain a time stamp, event identifier, and event severity.	incomplete	complete	complete
ACM004002	An event identifier must be encoded as a number, not a string, because strings consume too much of the limited downlink capacity.	complete	incomplete	complete
ACM004003	Elf must support three levels of event severity: a "green" level for purely informational events, a "yellow" level for warnings, and a "red" level for errors.	complete	complete	complete
ACM004004	Application programmers must be able to define new event types that contain application-specific data.	incomplete	incomplete	incomplete
ACM004005	The contents of an event object must be strongly typed so that downstream processing can access the contents in a type safe manner.	complete	incomplete	incomplete
ACM004006	Elf must define a signaling interface whereby an event occurrence is signaled with all the information needed to construct an event object.	incomplete	incomplete	incomplete
ACM004007	For reasons of runtime efficiency in high-performance applications, at least one of the signaling interfaces (if more than one) must be designed for speed in ignoring events for which logging is currently disabled.	complete	incomplete	complete
ACM004008	Elf must define interfaces for controlling entry policy and retention policy.	incomplete	incomplete	complete
ACM004009	It must be possible to change the tunable parameters of a policy at run-time, i.e, no source code changes and no recompilations.	incomplete	complete	incomplete
ACM004010	It is desirable to include with a logged event the source location where it was signaled.	complete	incomplete	incomplete
ACM004011	This helps distinguish between events that otherwise have the same signature.	incomplete	incomplete	incomplete
ACM004012	Since brevity is a virtue to most programmers, Elf should provide at least one signaling interface for basic events that can be written in a compact form.	incomplete	incomplete	complete
ACM004013	A programmer defines an application-specific event class.	complete	incomplete	incomplete
ACM004014	A programmer instruments source code to signal an event if it occurs.	complete	incomplete	complete
ACM004015	A running application program signals occurrence of an event.	incomplete	complete	complete
ACM004016	An Elf signaling mechanism checks the entry policy.	incomplete	incomplete	complete
ACM004017	Data management accepts an event object for logging.	incomplete	complete	complete

ACM004018	A ground program tabulates events as they arrive, sorting them by severity.	complete	incomplete	incomplete
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM004019	A ground operator adjusts the tunable parameters of a specific policy.	incomplete	incomplete	incomplete
ACM005001	There must be no limit on the number of securities traded (e.g., there are about 7000 securities listed at the SWX Swiss Exchange and about 10 times more at the German Stock Exchange).	complete	incomplete	complete
ACM005002	Securities must be groupable into segments.	incomplete	incomplete	complete
ACM005003	All trading parameters of the securities (price increments, minimum order sizes, stop trading ranges, trading hours, order book states) are kept in a database and must be configurable on-line on a per segment or a per security basis.	incomplete	complete	complete
ACM005004	Since the matching rules themselves change unfrequently, it is acceptable not to change them on-line, but to change them on source code level and to release a new version of OrderMatcher.	complete	incomplete	incomplete
ACM005005	The OrderMatcher must be extendible both in terms of orders matched per second and as far as the number of traded securities is concerned.	complete	complete	incomplete
ACM005006	A buy order with a price better than or equal to the incoming order price (from the point of view of the incoming order) is already in the order book.	complete	complete	complete
ACM005007	No order with a price better than the incoming order price (from the point of view of the incoming order) is already in the order book.	incomplete	complete	complete
ACM005008	A sell order with a price better than or equal to the incoming order price (from the point of view of the incoming order) is already in the order book.	complete	complete	complete
ACM005009	No order with a price better than the reference price (from the point of view of the incoming order) is already in the order book.	incomplete	complete	complete
ACM005010	No order with a price better than the arithmetic mean of the last two orders matched is in the order book.	incomplete	complete	complete
ACM005011	Buy order with a price better than the arithmetic mean of the last two orders matched is in the order book.	incomplete	complete	complete
ACM005012	The matching rules have to be adapted such that for the same price, customer orders have priority over orders entered on behalf of the broker, irrespective of their time of entry.	complete	complete	incomplete

KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM005013	The matching rules are adapted by the development department, tested with the members and put into operation on the pre-defined date.	complete	incomplete	complete
ACM006001	Users should be able to define and create a personal P2P network.	incomplete	complete	Complete
ACM006002	Desired properties and security measures can be specified while creating a network.	incomplete	complete	incomplete
ACM006003	Users must be able to instantiate and log in to existing or newly defined peer-to-peer networks or groups.	incomplete	complete	complete
ACM006004	Users must have the ability to define and implement security policies and measures relating to their own machine.	incomplete	complete	incomplete
ACM006005	Users can also receive a security analysis report of the network which lists the possible security hazards	incomplete	complete	complete
ACM006006	Online peers should be dynamically discovered.	complete	complete	incomplete
ACM006007	A List of available shared resources of the peers should be displayed upon request.	incomplete	complete	complete
ACM006008	A list of all online peers must be displayed dynamically on the user interface upon request	incomplete	complete	incomplete
ACM006009	A user should be able to send text messages to the connected peers.	incomplete	complete	complete
ACM006010	Instant messaging capability should support one-to-one and one-to-many modes.	complete	complete	complete
ACM006011	A user must be able to transfer files to a selected connected peer.	incomplete	complete	complete
ACM006012	A user must be able to utilize the computing resources such as processing power, memory, and cache of the connected peers to perform computing resource consuming tasks.	incomplete	incomplete	incomplete
ACM006013	The user can specify either serial or parallel execution of the sub-tasks on peer machines.	incomplete	incomplete	complete
ACM006014	Users should be able to use the intelligent search agents for performing a variety of tasks over the P2P network.	incomplete	complete	incomplete
ACM006015	These agents should be configurable to specify the search criteria and also the actions that the user wants the agent to perform on their behalf.	incomplete	complete	incomplete
ACM006016	Users must be informed of any errors that might have occurred during file transfer or messaging	incomplete	complete	complete
ACM006017	Proper logs should be generated for any resource use by the peers.	complete	incomplete	complete
ACM006018	Possible security hazards must be forecasted in the logs as well as user prompts.	complete	incomplete	incomplete

KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM007001	The bidding system should be able to let the customer send the request for quotation.	complete	complete	complete
ACM007002	A detailed description form should be provided by the system.	incomplete	complete	complete
ACM007003	After the calculation, the quotation can be sent to the customer.	incomplete	complete	complete
ACM007004	The customer should be able to change requirements about the project as necessary by using the system.	complete	incomplete	incomplete
ACM007005	The bidding system should enable users to enter customer information into the database	incomplete	complete	incomplete
ACM007006	According to the detailed description of project, the bidding system should be able to connect to various vendors' public databases to request material pricing (if material is required for the RFQ).	incomplete	incomplete	complete
ACM007007	According to the detailed project description, the bidding system should be able to connect to the company's internal database to calculate the labor and other costs associated with a particular RFQ.	incomplete	incomplete	complete
ACM007008	The system should provide means for transmitting a fully compliant response to the RFQ (a Quotation), back to the customer.	incomplete	complete	incomplete
ACM007009	If materials required in the RFQ specification details are not available, the system should be able to send a message about alternative materials to the customer, or to request clarification on other or unclear portions of the RFQ specification.	incomplete	complete	complete
ACM007010	The system should be able to calculate the total cost of the project and produce a formal, and fully compliant quotation, as a response to the customer RFQ.	incomplete	complete	incomplete
ACM007011	The bidding system should also be able to produce separate pricing details for each job site (if the project includes more than one site).	complete	complete	complete
ACM007012	The system should be able to search archived records and provide details and cost data of similar previous projects, including unsuccessful bids as well as completed projects.	complete	complete	complete
ACM007013	The system will locate the best price from the different vendors.	incomplete	complete	complete
ACM007014	If the RFQ detail specifications change in any material way prior to the bid date, the system will provide a new quotation reflecting all such changes.	incomplete	complete	complete
ACM007015	New customers register with the company, and provide detailed information, for example, contact information.	complete	incomplete	complete

ACM007016	Customer may add, change, or delete its profile information as necessary.	complete	incomplete	complete
ACM007017	A registered user uses the web interface to fill the RFQ form.	incomplete	incomplete	complete
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM007018	A formal addendum process is often involved with most large RFQs.	complete	incomplete	incomplete
ACM007019	The Customer sends Addendum in response to questions or clarifications posed by the company or its subcontractors and thought necessary to the specifications.	incomplete	incomplete	complete
ACM007020	A formal addendum process always involved with most large RFQs.	complete	incomplete	complete
ACM008001	The user should be able to post a requirement posting on the system for its other users that are the suppliers, dealers or the manufacturers.	incomplete	complete	incomplete
ACM008002	The system should provide an interface for entering users' preferences	incomplete	complete	complete
ACM008003	The system should be able to classify and categorize the various needs of buyers and post them to the corresponding suppliers, helping the suppliers to view the details of the initiated posting by a particular buyer	incomplete	complete	complete
ACM008004	User should have the option of making his/her posting public, private or say for a particular dealer or manufacturer.	complete	complete	incomplete
ACM008005	The user should be able to provide the time frame of the reverse auction.	incomplete	complete	complete
ACM008006	The user should have the facility to view the final result categorized and customized according to his requirement.	incomplete	complete	incomplete
ACM008007	The final verdict remains with the buyer whether to choose a particular offer or not.	incomplete	complete	complete
ACM008008	A user can post his request for a car by specifying his requirement including the manufacturer's company, the model of the car, the year of Make (if opting for a used car), transmission type, preference of buying, target price and so on.	complete	incomplete	incomplete
ACM008009	The system classifies and categorizes the requests posted by buyers and posts them to the corresponding suppliers, dealers, or owners.	complete	complete	complete
ACM008010	The system sets a time frame (provided by buyer) that limits the time for bidding for a specific posting.	incomplete	complete	incomplete
ACM008011	The buyer views all the biddings posted to his account.	incomplete	complete	incomplete
ACM008012	The user (buyer/seller) follows the hyperlink to view the details of the corresponding seller/buyer.	complete	complete	incomplete
ACM008013	The User specifies the preferences of the car that is desired like the model, year and other parameter desired.	complete	complete	incomplete

ACM008014	The user sets the request starting date and request ending date for bidding.	incomplete	complete	complete
ACM008015	The user specifies the approximate price desired.	incomplete	complete	complete
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM008016	The user selects the type of sellers that should be shown the request (Optional).	incomplete	complete	complete
ACM008017	The user confirms the specifications to be posted.	incomplete	complete	complete
ACM008018	The UserBuyingRequest Class provides the specifications of the buying request to the RequestHandler.	complete	complete	complete
ACM008019	Search for the corresponding users (sellers) according to preferences.	incomplete	complete	incomplete
ACM008020	Specifications and preferences provided by the user (buyer) are compared with the posted available car(s) and/or future/potential availability.	complete	complete	complete
ACM008021	If the posting preferences match the details of seller, the request is posted to the corresponding seller.	incomplete	complete	incomplete
ACM008022	RequestHandler class sends information to AccountUpdater Class that updates the account of both the prospective seller and prospective buyer	complete	complete	complete
ACM008023	The user is identified as the buyer	incomplete	complete	complete
ACM008024	The user Id is searched in the buying requests for corresponding match in the buyer field of the User Buying Request.	complete	complete	complete
ACM008025	Present the output to the buyer in a tabularized format that has all the buying requests that he has initiated.	incomplete	incomplete	incomplete
ACM008026	The user (seller/buyer) requests for details of the buying request	incomplete	complete	complete
ACM008027	All the requests corresponding to this user are shown in a tabularized form with hyperlink to get the details of the same.	complete	incomplete	incomplete
ACM008028	The user clicks on the hyperlink to get the detailed information about the request.	incomplete	incomplete	complete
ACM008029	All the information (data and current status) is shown to the user.	incomplete	complete	incomplete
ACM008030	The user selects to bid for a particular request posted in his account.	complete	complete	incomplete
ACM008031	User (seller in this case) quotes for all the required fields as price, quantity, availability time period etc.	complete	incomplete	incomplete
ACM008032	The BidresponseHandler checks for price to be lower than any previous posted price (Refer Use Case 5-0) and provides Error Checking.	complete	incomplete	complete
ACM008033	BidResponseHandler updates the account of the bid sender and the initiator.	incomplete	complete	complete
ACM008034	The RequestStatus class reads all the bids posted for the particular buying request.	complete	complete	complete

ACM008035	RequestStatus class gathers the priceQuoted from the bid.	incomplete	complete	complete
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM008036	It compares the price quoted with a large number and recursively compare it with other prices for the lowest price.	incomplete	incomplete	incomplete
ACM008037	The prices are arranged in an ascending order.	complete	complete	complete
ACM008038	Provide the Value of the current lowest bid.	incomplete	complete	incomplete
ACM008039	The user Id is identified as the Potential Seller	complete	complete	complete
ACM008040	All the bids in which the user is identified as potential seller are selected and those alive are finally chosen.	complete	complete	complete
ACM008041	Present the output (bids posted) to the user in a tabularized and ascending order format with regard to time.	complete	incomplete	incomplete
ACM008042	A comparison is made for the buying Request about its status for checking whether it is still active or not.	complete	complete	incomplete
ACM008043	If alive, for a particular request the current time is subtracted from the sum of the initiating time and the duration given by the potential buyer at the time of initiating the buying request.	incomplete	incomplete	complete
ACM008044	The time remaining once calculated is used to provide screen alerts for the user Account.	complete	complete	complete
ACM008045	The user (buyer/seller) clicks on the hyper link of the seller/buyer name/id.	complete	incomplete	complete
ACM008046	The user Id is picked up and searched for.	incomplete	complete	incomplete
ACM008047	The details of the user and selected and presented in the corresponding HTML Template.	incomplete	incomplete	incomplete
ACM008048	Show the corresponding template to the user.	incomplete	incomplete	complete
ACM008049	The potential buyer clicks on the enquire by e mail button.	incomplete	complete	complete
ACM008050	A HTML screen template is presented with the field of outgoing mail populated with the user Id of the seller.	complete	complete	complete
ACM008051	The potential buyer fills in the Text area for the inquiry and also particular comments if any and press send button for sending the message.	complete	complete	complete
ACM008052	An e mail inquiry is posted to the corresponding seller's account.	incomplete	complete	complete
ACM008053	Time Remaining is calculated for the corresponding user request by use case 5-3.	complete	complete	complete
ACM008054	If the time elapsed is greater than or equal to the bid duration time the bid is marked	complete	complete	complete

	for closing (free zed).			
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM008055	If a user tries to access the same or bid for it the system gives an error message saying that the bid period is already over for this request.	incomplete	complete	complete
ACM009001	The system should support rental and check in operations and provide pricing of the rental items.	incomplete	complete	complete
ACM009002	It should track member financial activity and track current rentals outstanding and over due items held by members.	incomplete	complete	incomplete
ACM009003	The system should track daily income, video returns, and overdue videos, at minimum.	complete	complete	incomplete
ACM009004	The system should grow gracefully to handle multiple stores.	incomplete	incomplete	incomplete
ACM009005	Automate check in and check out of rental items, handle logging and reporting of rental transactions	complete	complete	complete
ACM009006	Provide real-time financial information.	incomplete	complete	incomplete
ACM009007	Provide flexibility in quickly implementing pricing plans and promotions.	incomplete	incomplete	incomplete
ACM009008	Support company growth to multiple stores.	incomplete	incomplete	incomplete
ACM009009	The sales clerks answer customer questions and check out the videos and other items the customers are renting.	incomplete	complete	incomplete
ACM009010	They are also responsible for making calls about overdue videos.	incomplete	complete	incomplete
ACM009011	The clerk presses the "Rent" button and scans, or enters the item ID (by scanning the bar code or entering the bar code number) into the screen.	complete	complete	complete
ACM009012	Rentals will only be made to people who are signed up as members	complete	complete	complete
ACM009013	People will only be accepted as members who have provided adequate identity and credit information.	incomplete	complete	incomplete
ACM009014	Rentals will be denied to anyone with an outstanding balance of \$20 or more.	complete	complete	complete
ACM009015	Only a manager may change pricing or member policy	incomplete	complete	complete
ACM009016	Using the system must be at least as fast as manual checkouts.	incomplete	incomplete	incomplete
ACM009017	Point of sale unit hardware must be Java enabled.	complete	complete	complete
ACM010001	The mechanism must cause no more changes than shown above.	incomplete	incomplete	incomplete
ACM010002	The mechanism should be defined as a simple set of rules that the subsystems must follow, and at the same time, that are clear enough to lead directly to the design of	incomplete	complete	incomplete

	AM and the new control interfaces of subsystems.			
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM010003	The mechanism should be extensible to N-subsystems, (future-proof, when we decide to integrate additional existing appliance, e.g Super Audio CD).	complete	incomplete	complete
ACM010004	The mechanism should be flexible enough to accommodate changes in scenarios (because of changes in requirements or because of late bugs in scenarios).	complete	complete	incomplete
ACM011001	The software should select the right detector mode, requested by the control system of the X-ray modality, as well as retrieve the right set of calibrations needed for the Image Processing.	complete	complete	complete
ACM011002	The detector needs to be read-out with a fixed frequency.	incomplete	incomplete	incomplete
ACM011003	Mode changes have to be synchronized with the read-out request: A 'Change Mode' command will be send during read-out, after which the detector will switch to the new mode.	complete	complete	complete
ACM012001	Each member's latest login date and time are displayed in this section.	complete	complete	complete
ACM012002	Each member who is logged on to the family can see a list of other family members logged in at the same time.	complete	complete	complete
ACM012003	The members can change their passwords at any time using the Change Password button	incomplete	complete	complete
ACM012004	Each member can post short messages (less than 100 words) on the bulletin board of the family.	complete	complete	complete
ACM012005	All members who log on after the message is posted can view this message and reply to it if they want.	complete	complete	complete
ACM012006	Each member can upload photos to the webpage to be viewed by all other members.	complete	complete	complete
ACM012007	Every member's name, address, telephone numbers and email ids are stored in the address book, to be viewed by other members.	complete	complete	complete
ACM012008	A logged on member can send emails to anyone in the address book, anyone whose details are not in the address book, and group emails to the entire group.	incomplete	complete	complete
ACM012009	The calendar contains birthdays of all members who have logged on and entered this information.	complete	complete	incomplete
ACM012010	Features like 'Chat' & 'Conference' enable the members of the family to communicate with each other.	complete	complete	complete

KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM012011	They can either chat with individual members of the group or they can chat in a 'Conference' with more than one member.	incomplete	complete	complete
ACM012012	The user can decide who he wants to chat with from the list of currently logged on members.	incomplete	complete	incomplete
ACM012013	Only the owner of the group can add members to the group.	Complete	complete	complete
ACM012014	The user goes to www.MyFamilyReunion.com to create a new group/family	Complete	complete	complete
ACM012015	Only the owner of the group can add members to the group.	Complete	complete	complete
ACM012016	A person logging on for the first time goes to www.MyFamilyReunion.com and selects the family that he wishes to log on to.	incomplete	complete	incomplete
ACM012017	To post a message, the member clicks on the post message link.	incomplete	complete	complete
ACM012018	The users can also reply to a posted message.	incomplete	complete	complete
ACM012019	The member clicks on the 'Photos' link and is taken to a page where there are thumbnail images of all the photos that have been uploaded thus far.	Complete	complete	complete
ACM012020	From the family homepage, the user is taken to the address book page by clicking on the 'Addresses' link.	incomplete	complete	complete
ACM012021	Users can send mails to the other members in the group by selecting their mail addresses from the Address Book.	incomplete	complete	complete
ACM012022	A list of members, who are currently logged on, is displayed in a column labeled as 'Online'.	Complete	complete	complete
ACM012023	This feature enables more than one online member to chat in a group.	incomplete	complete	incomplete
ACM013001	An administrator adds a new therapist.	Complete	complete	complete
ACM013002	Essential personal data of each therapist must be kept in the system, like name, address, telephone number, email address.	Complete	incomplete	incomplete
ACM013003	An administrator adds a user to the system who is a parent.	incomplete	complete	complete
ACM013004	An administrator adds a user to the system who is a doctor.	incomplete	complete	complete
ACM013005	A therapist logs on and adds a new patient	Complete	complete	complete
ACM013006	The therapist adds the essential data of the child and parents, and assigns him/herself as the primary responsible therapist.	Complete	complete	incomplete
ACM013007	Treatment plans and first appointments are also added.	incomplete	complete	incomplete

KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM013008	A therapist adds a new appointment or changes an existing one.	incomplete	complete	complete
ACM013009	When the therapist has logged- in and asks to access the appointment system, a calendar is shown with an overview of existing appointments.	Complete	complete	incomplete
ACM013010	The therapist can select an empty time slot and enter new data, or he/she can select an already taken time slot and change the contents.	Complete	complete	incomplete
ACM013011	A therapist enters goals, a treatment plan, or uploads a progress report.	incomplete	complete	complete
ACM013012	The system presents specific forms for each item that the therapist wants to keep.	incomplete	complete	incomplete
ACM013013	A therapist assigns a replacement therapist for a patient.	incomplete	complete	complete
ACM013014	A parent accesses the system to review the progress of a child.	incomplete	complete	complete
ACM013015	A doctor accesses the system to review the progress of a child.	incomplete	complete	complete
ACM013016	The system knows for which children this user can access data and retrieves the requested reports.	complete	complete	complete
ACM013017	Search and show the appointment history for a particular child.	incomplete	complete	incomplete
ACM014001	A customer must be created before a contract can be created for her.	incomplete	complete	incomplete
ACM014002	A customer can have zero, one or more that one contract.	incomplete	complete	complete
ACM014003	The legacy system has an API that returns a random available phone number, or will return a list of available phone numbers given an area code and prefix.	complete	complete	complete
ACM014004	All services provided by the Service Tier can be called individually by the Presentation Tier.	complete	complete	incomplete
ACM014005	All services that will be provided by the Service Tier take some number of parameters.	complete	incomplete	incomplete
ACM014006	The services in a set must be executed in a particular order defined by the set (an ordered set).	complete	incomplete	incomplete
ACM014007	Individual services are defined at compile time.	incomplete	complete	incomplete
ACM014008	It must be easy for the Service Tier programmers to add new individual services.	incomplete	complete	incomplete
ACM014009	Sets of services have optional and execute attributes the same as individual services.	incomplete	complete	incomplete
ACM014010	The application starts with its parameter file that defines a set of services that includes Address Validation, Black List Validation, and Create Customer, in that order.	complete	complete	incomplete
ACM014011	The application creates a customer with a bad address.	incomplete	complete	complete

ACM014012	The application searches for the customer just created and displays it.	incomplete	incomplete	incomplete
ACM014013	After finding and display the customer, a contract is created and added to the customer.	complete	complete	incomplete
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM014014	The customer or contract information displayed and updating the system with these changes.	incomplete	complete	incomplete
ACM014015	The application has an GUI that allows the use to turn Address Validation and BlackList Validation on or off by using check boxes.	complete	complete	complete
ACM014016	The application starts with its parameter file that defines a set of validation services.	incomplete	incomplete	incomplete
ACM014017	The application creates a customer with a bad address.	incomplete	complete	complete
ACM014018	The Presentation Tier is informed of the Address Validation failure as well as the success of Black List Validation (the customer name and SSN is not in the Black List database) and the customer is created.	complete	complete	incomplete
ACM015001	The client application (in this case, a browser) sends a request to Oracle Web Application Server.	complete	complete	complete
ACM015002	The output file (completed contract document in PDF format) is put into a designated output folder.	complete	complete	complete
ACM015003	Once the output file appears in the output folder, the component unblocks itself.	complete	incomplete	complete
ACM015004	The Application Server then returns the URL of the completed lease (PDF file) and the return code to the calling application (in this case a browser).	complete	complete	complete
ACM015005	When the user clicks on the link on the client browser , the PDF file will be downloaded to the client machine.	complete	complete	complete
ACM015006	An organisation using a standard web browser requests a lease contract over the Internet.	complete	complete	incomplete
ACM015007	A blank document is requested.	complete	complete	incomplete
ACM015008	The Web server forwards the contract request to the NDDS.	complete	complete	complete
ACM015009	When the client gets the document back at the browser, he/she prints out a hard copy and has it signed by the customer.	complete	complete	complete
ACM015010	The original signed documents are faxed back to the company for archiving as legal contracts.	complete	complete	complete
ACM015011	An organisation using a standard web browser requests a lease contract over the	complete	complete	incomplete

	Internet.			
KODE	Pernyataan Kebutuhan	Ahli 1	Ahli 2	Ahli 3
ACM029011	Allow users to change server side spam filter options.	complete	complete	complete
ACM029012	Serve a order form for users to alter their service level.	incomplete	complete	complete
ACM030001	The system will work in a proper speed so the required time for study will be decreased.	incomplete	incomplete	incomplete
ACM030002	When the doctor is able to see the different places of the organ from various points of view, the chance of mistakes will be reduced.	incomplete	incomplete	complete
ACM030003	The system provides doctors the opportunity to study the tissues precisely.	incomplete	incomplete	complete
ACM030004	The coloring and contrast adjustment helps them concentrate on a specific group of cells.	incomplete	complete	incomplete
ACM030005	The system can save the results.	complete	incomplete	complete
ACM030006	The MRI Visual Analyzer is easy to use.	incomplete	incomplete	incomplete
ACM030007	Doctors and hospital personals that are not necessarily computer experts will use MRI Visual system.	complete	incomplete	complete
ACM030008	The system shall have file transportation options such as open, close, save, save as, print, send to, and exit.	complete	incomplete	complete
ACM030009	Doctors should be able to do any process in the system both by clicking on the proper icon and by using the hot keys.	complete	incomplete	complete
ACM030010	The results of measurements should be available in both metric and British systems.	complete	incomplete	complete

Petunjuk Annotator Pembuatan *Corpus Complete* dan *Incomplete* Pada Pernyataan Kebutuhan Perangkat Lunak

1. Tujuan

Tujuan dalam melakukan anotasi adalah mengklasifikasikan pernyataan kebutuhan yang lengkap dan tidak sesuai dengan kepakaran ahli yang selanjutnya digunakan sebagai dataset penelitian.

2. Definisi Karakteristik Lengkap (*complete*) pada Pernyataan Kebutuhan Perangkat Lunak

Sebuah pernyataan kebutuhan yang lengkap menunjukkan setiap pernyataan kebutuhan dapat dipahami tanpa melibatkan atau terkait pada kebutuhan lain. Pernyataan kebutuhan seharusnya kalimat lengkap yang tidak memerlukan refesensi pada pernyataan lain untuk dipahami sebagai bentuk dasar (INCOSE, 2015).

Definisi lain juga menyatakan sebuah kebutuhan dikatakan lengkap jika terdiri dari semua informasi yang terhindar dari *ambiguity* dan tidak perlu penjelasan tambahan (Firesmith 2005).

Beberapa contoh-contoh pernyataan kebutuhan yang lengkap dan tidak lengkap:

Contoh 1 :

- 1) The controller shall send the driver his itinerary for the day.
- 2) The controller shall send the driver_itinerary for the day to the Driver.

Pada kalimat 1) merupakan contoh kalimat tidak lengkap karena terdapat kata ganti “his” yang masih memerlukan penjelasan lain siapa his tersebut. Pada kalimat 2) merupakan bentuk kalimat yang lengkap dengan mengganti kata ganti *his* dengan kata benda.

Contoh 2 :

- 1) The Flight Information System shall usually be on line.
- 2) The Flight Information System shall have an availability of at least xx% over a period of at least yy hours.

Pada kalimat 1) merupakan contoh kalimat tidak lengkap karena terdapat bentuk ekspresi subjektif “*usually*” yang masih belum menjelaskan keterangan waktu secara tepat. Pada kalimat 2) merupakan bentuk kalimat yang lengkap dengan mengganti kata *usually* yang dijelaskan lebih rinci.

Contoh 3 :

- 1) The Circuit_Board shall have a storage temperature or not more than 30 degrees.
- 2) The Circuit_Board shall have a storage temperature or not more than 30 degrees Celsius.

Kalimat 1) merupakan contoh kalimat tidak lengkap karena tidak menjelaskan satuan derajat secara jelas. Dan kalimat 2) merupakan bentuk kalimat yang lengkap dengan memberikan satuan derajat celcius terhadap temperatur

Contoh 4 :

- 1) The Flight_Information_System shall display the Tracking_Information for relevant aircraft.
- 2) The Flight_Information_System shall display the Tracking_Information of each Aircraft located less than or equal to 20 kilometers from the Airfield.

Pada kalimat 1) merupakan contoh kalimat tidak lengkap karena terdapat bentuk ekspresi subjektif “*relevant*” yang masih belum menjelaskan kondisi secara jelas. Pada kalimat 2) merupakan bentuk kalimat yang lengkap dengan mengganti kata *relevant* yang dijelaskan lebih rinci .

Contoh 5 :

- 1) The ATM shall display the Customer Account_Number, Account_Balance, and so on.
- 2) (Split into as many requirements as necessary to be complete):
The ATM shall display the Customer Account_Number.
The ATM shall display the Customer Account_Balance.
The ATM shall display the Customer Account_Type.

The ATM shall display the Customer Account_Overdraft_limit.

The ATM shall display the Customer

Kalimat 1) merupakan contoh kalimat tidak lengkap. Dan kalimat 2) merupakan bentuk kalimat yang lengkap dengan merinci kebutuhan secara jelas.

3. Langkah-langkah :

1. Membuka file Corpus.xlsx yang berisi daftar kalimat kebutuhan.
2. Membaca serta memahami setiap kalimat kebutuhan
3. Menentukan apakah kalimat tersebut lengkap(*complete*) atau tidak lengkap (*incomplete*)
4. Menulis *complete* atau *incomplete* pada kolom klasifikasi.

BIODATA PENULIS



Penulis, **Suci Nurfauziah**, lahir di Lamongan, 14 April 1990. Biasa dipanggil dengan nama Suci. Anak pertama dari 2 bersaudara dan dibesarkan di kabupaten Lamongan, Jawa Timur. Penulis menempuh pendidikan formal di SDN Tlogoanyar (1996-2002), SMP Negeri I Magetan (2002-2005), SMA Negeri I Magetan (2005-2008) Pada tahun 2008-2013 penulis melanjutkan studinya di jurusan Teknik

Informatika, Fakultas Sains dan Teknologi, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pada tahun 2015-2017, penulis melanjutkan pendidikan Magister S2 di jurusan yang sama, yaitu Jurusan Teknik Informatika, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya, Jawa Timur. Di Jurusan Teknik Informatika, penulis mengambil bidang minat Rekayasa Perangkat Lunak. Penulis juga pernah bekerja di PT. Geomedia sebagai analis pada tahun 2012-2014. Selain itu juga, penulis juga pernah mengajar di Universitas Hasyim Asy'ari Jombang selama menempuh perkuliahan tahun 2014-2016. Penulis dapat dihubungi melalui alamat email sucinurf144@gmail.com.